Using Git/GitHub in Scientific Collaborations

Stockholm 2021-01-26

Mikkel C. Vinding, PhD
NatMEG, CNS
Karolinska Institutet

Disclaimer: I have no affiliation with Git or GitHub, Inc.

Overview

- What is Git/GitHub?
- Starting a GitHub project
- Project workflow
- Collaborating
- Tutorial

```
$ git pull origin master
```

Code example

GitHub for (neuro)scientific collaboration

Why?

1. Organize your data analysis scripts
2. Easy to collaborate on tasks that require programming (i.e. all neuroscientific data analysis)
3. Share finished analysis scripts

Solution: use Git and GitHub

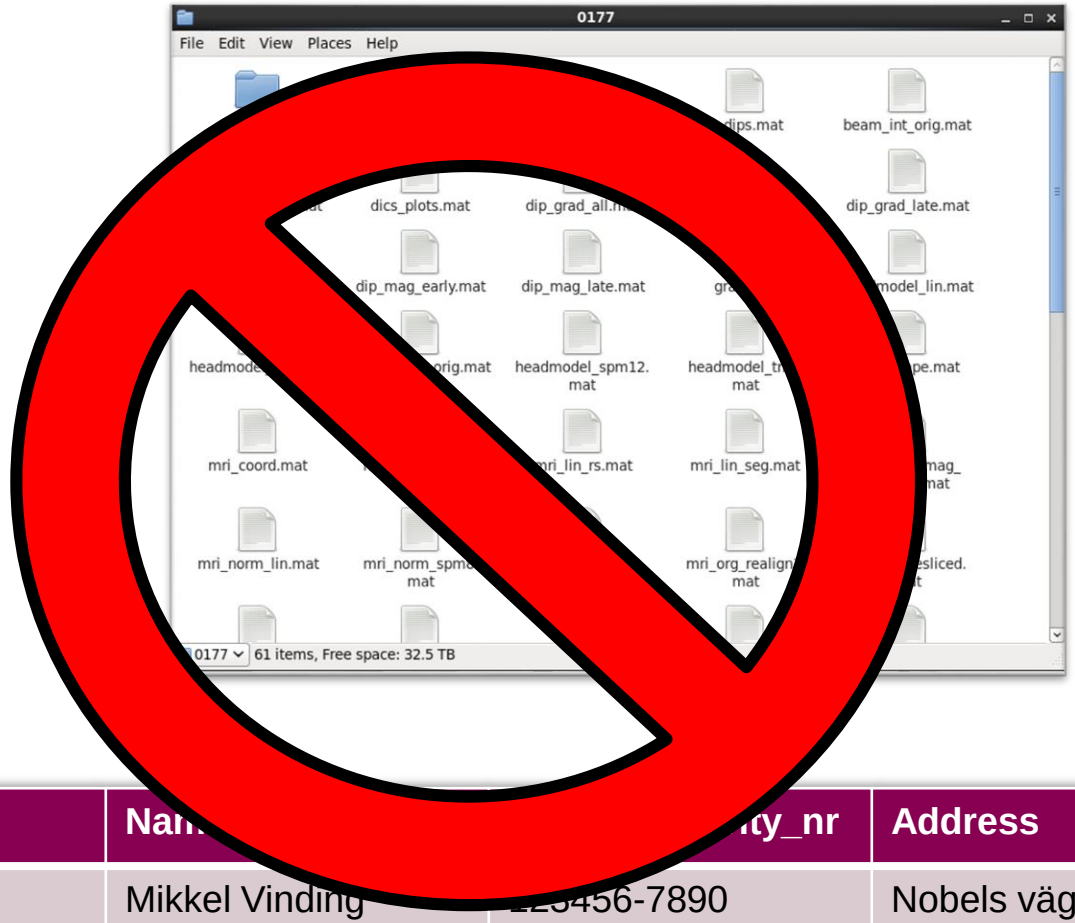
What is Git and GitHub?

- Tool for distributed version control
- Free and open source
- https://git-scm.com/

- Hosting service and interface for Git
- Owned by Microsoft, but free*
- Options for project management, collaboration, wikis , and more
- https://GitHub.com/

What NOT to put on GitHub

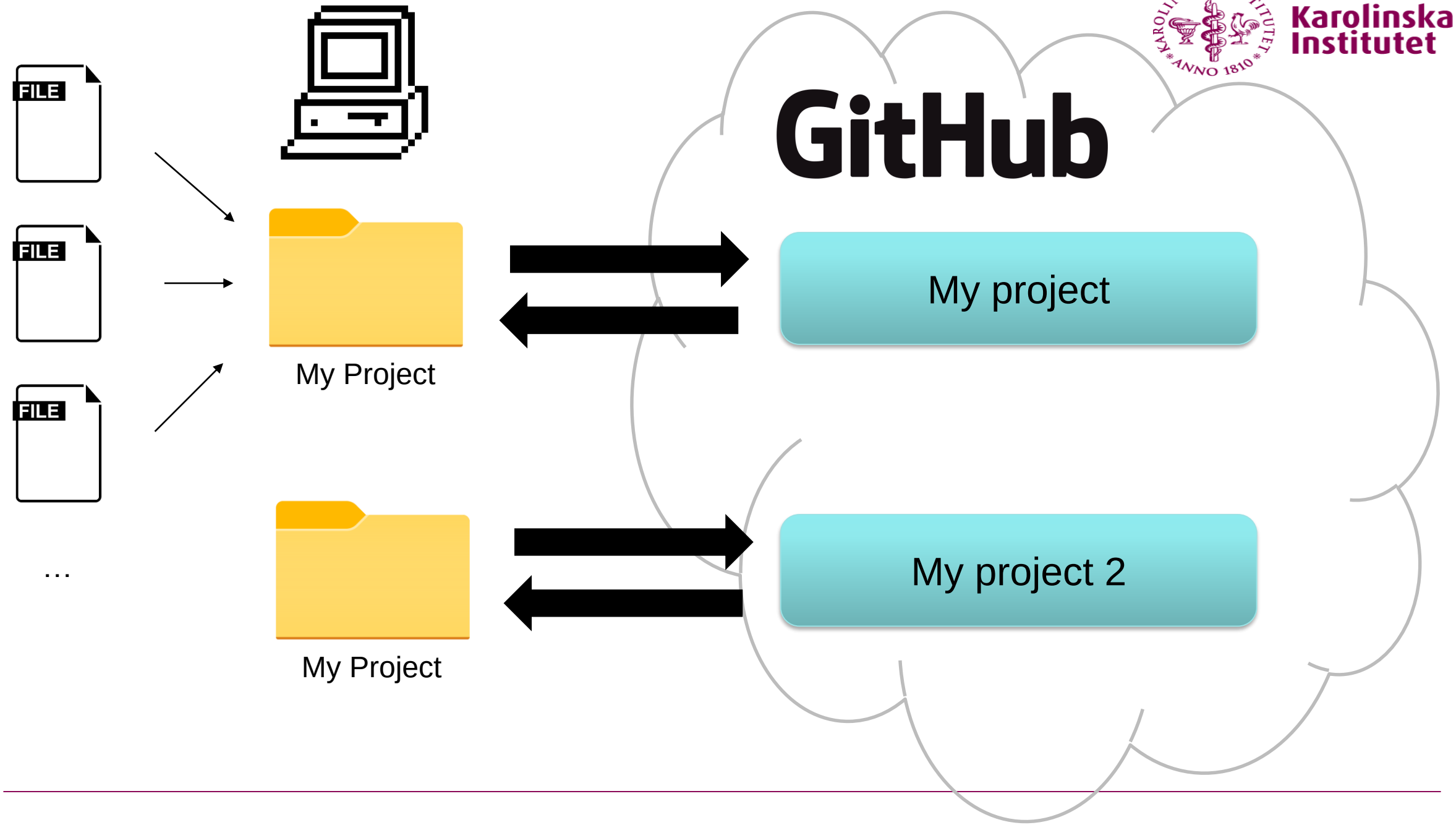
- Datafiles
  - →Maximum file size is 100MB
  - →Maximum repository size is 10GB
- Sensitive information



Subjid	Nam	ty_nr	Address
0001	Mikkel Vinding	123456-7890	Nobels väg 9
0002	…		

Karolinska Institutet

GIT WORKFLOW

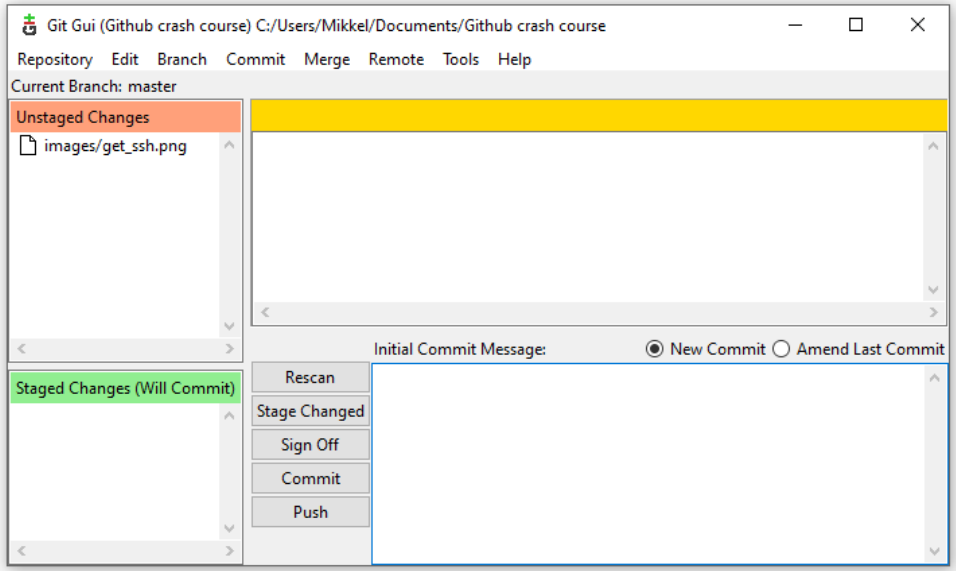

FILE
FILE
FILE
...
My Project
My Project
GitHub
My project
My project 2

Using Git

```
$ git <command> <option(s)> <files> <…>
```

Do I have to use the terminal?

Yes*



*No: There is a GUI

Karolinska Institutet

START A GIT PROJECT

Getting started…

Start a new project

Create a repository for your project.

Where you want to start!

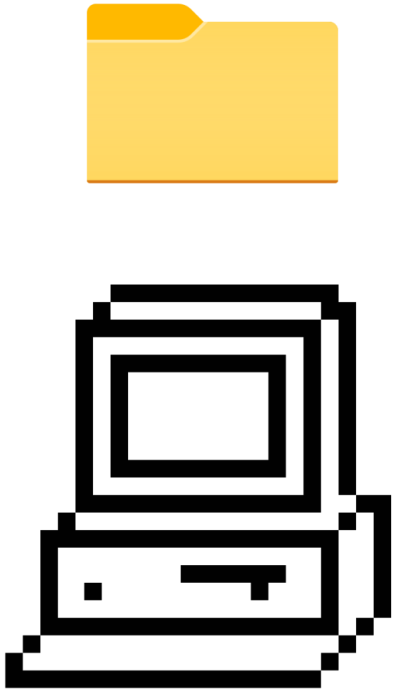
Join a project

E.g. a project where someone already hace created a repository

Clone a project

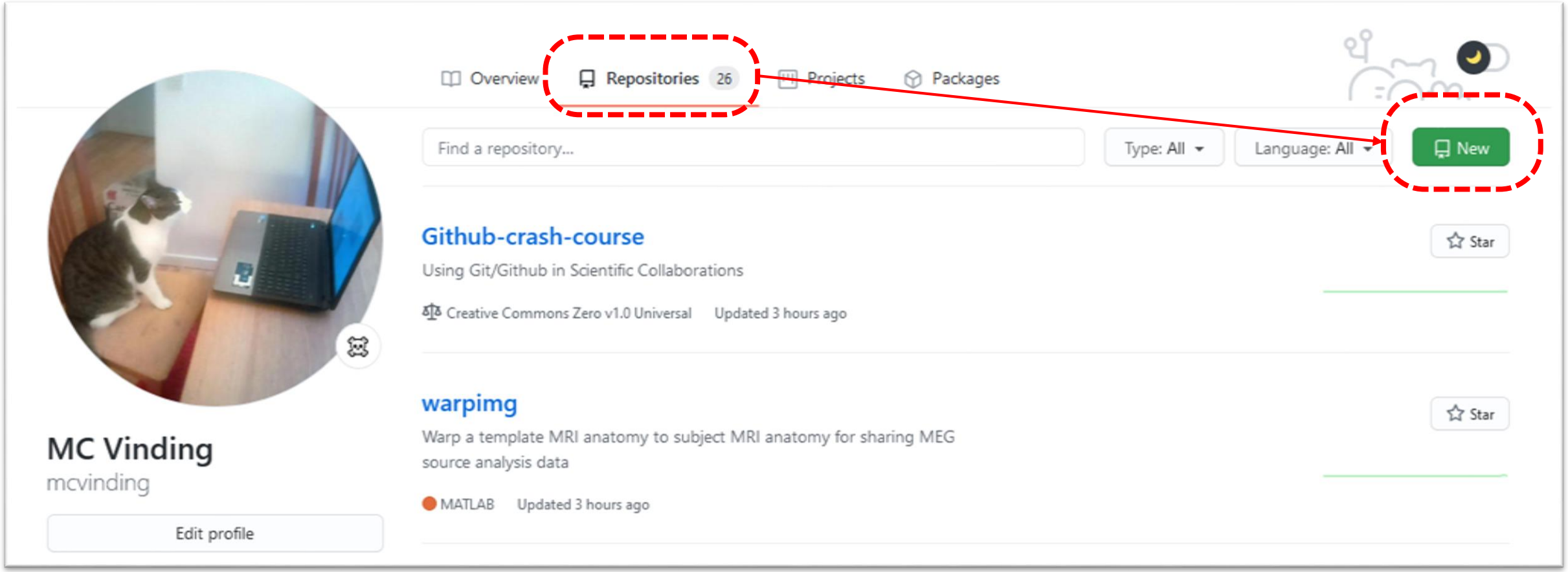
Get code from another project, e.g. an analysis toolbox

Start a new project…



GitHub

Start a new project…

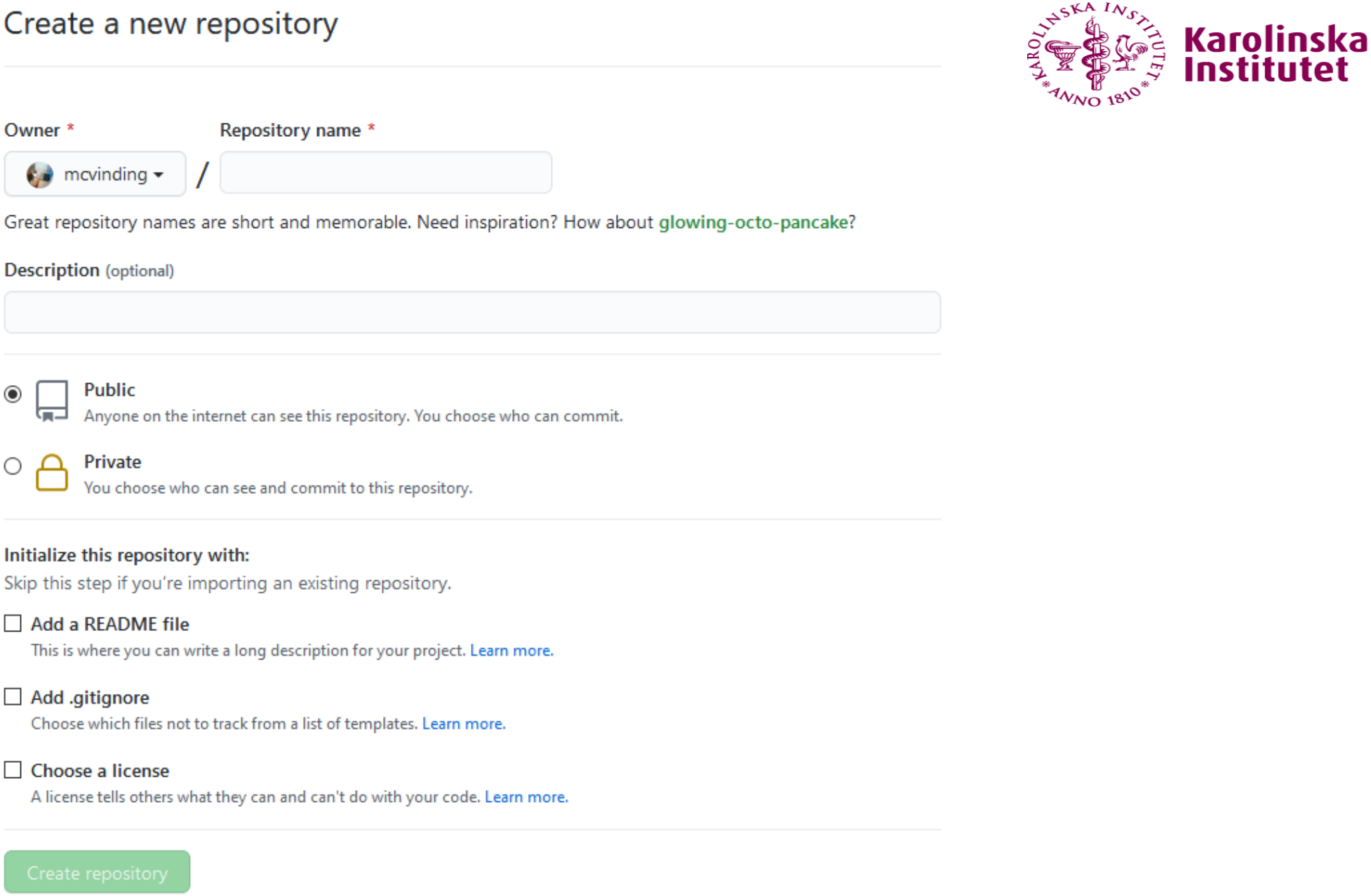


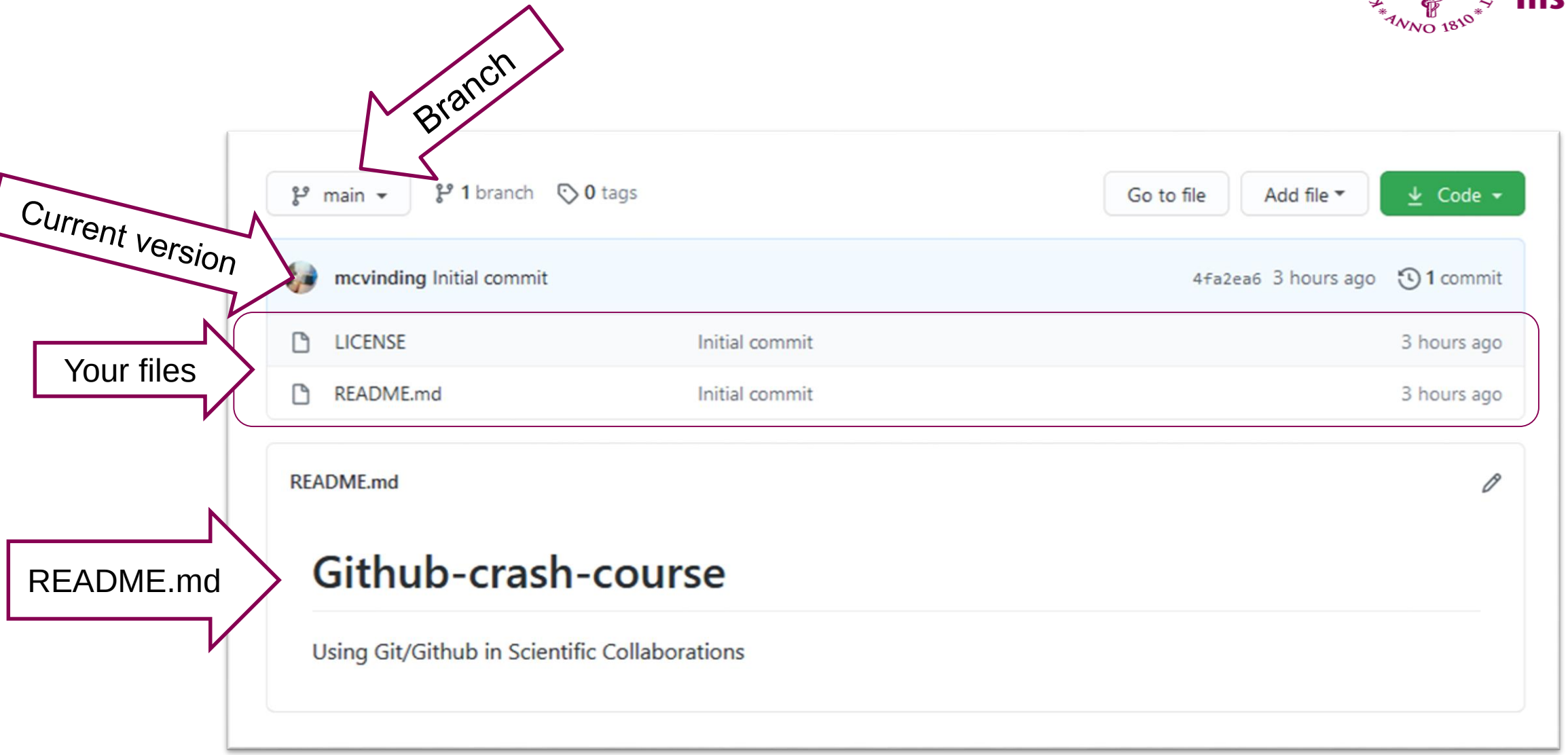
1. Name

2. Define visibility

3. ????

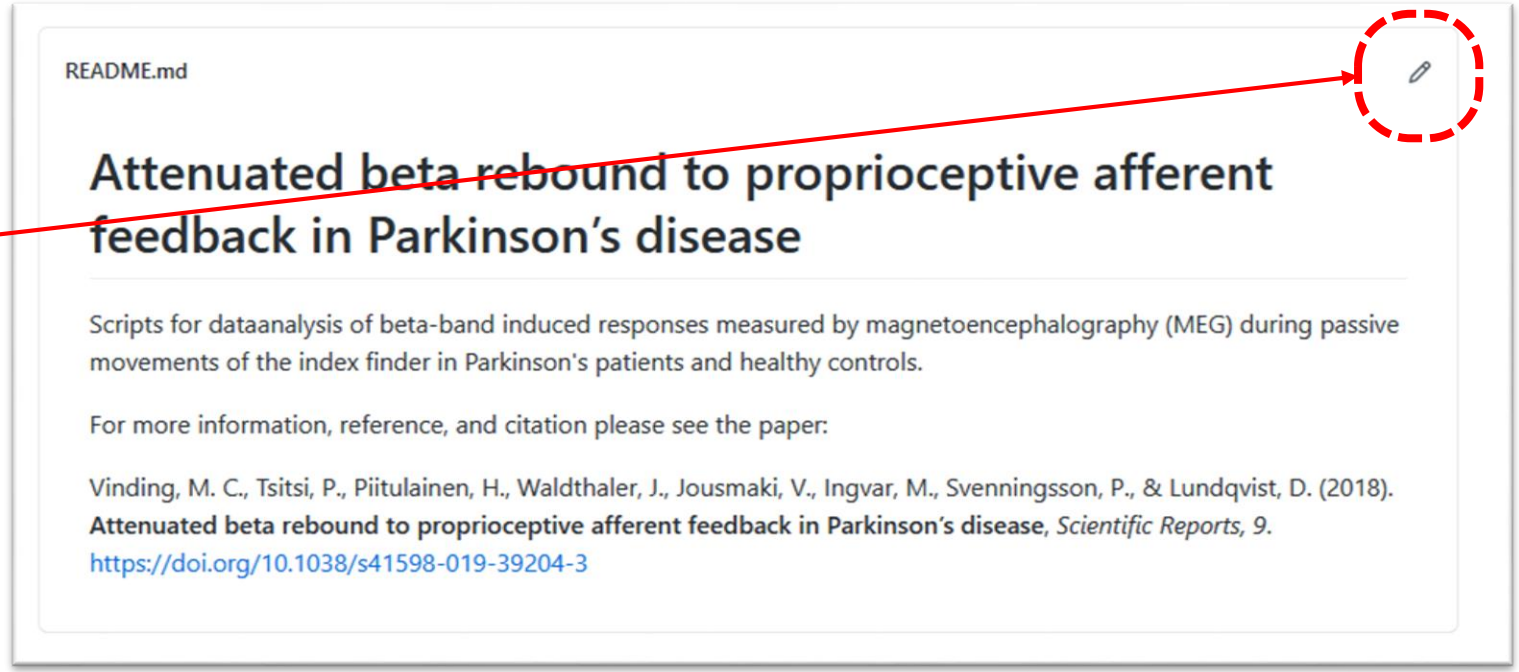
4. Create




Branch
Current version
Your files
README.md
main
1 branch
0 tags
Go to file
Add file
Code
mcvinding Initial commit
4fa2ea6 3 hours ago
1 commit
LICENSE
Initial commit
3 hours ago
README.md
Initial commit
3 hours ago
README.md
Github-crash-course
Using Git/Github in Scientific Collaborations

The README file

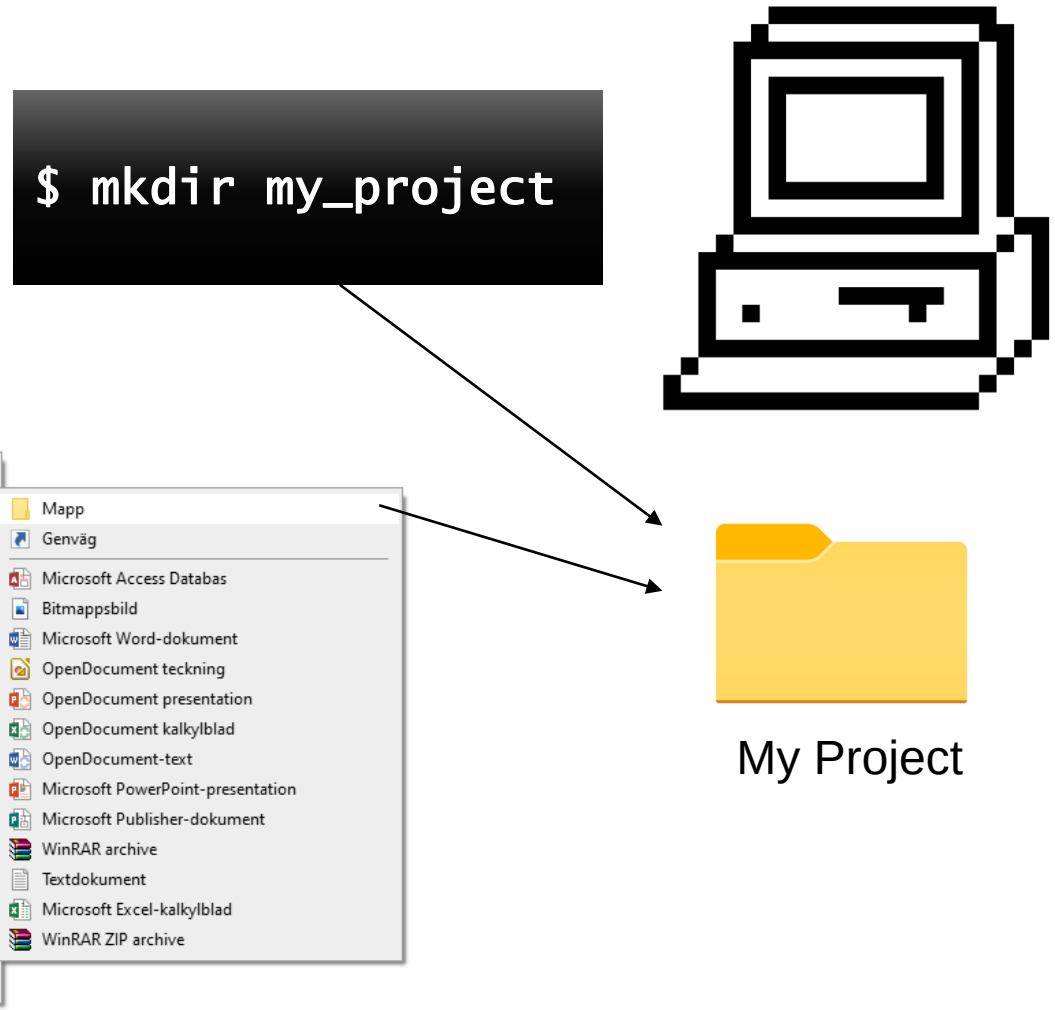
- **Write *informative text* that help people who find your repository**
  - → Collaborators…
  - → Reviewers…
  - → Public…
- Edit in text editor or in browser
- Markdown syntax



Nice Markdown cheatsheet: https://GitHub.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

Start a new project…

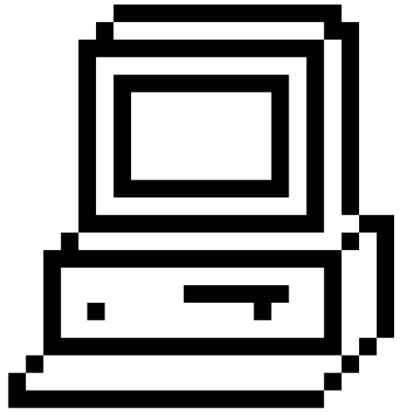
- Create a folder at the desired location



Start a new project…

- Create a folder at the desired location
- Go to folder

```
$ cd my_project
```



My Project

Start a new project…

- Create a folder at the desired location
- Go to folder
- Initialize folder

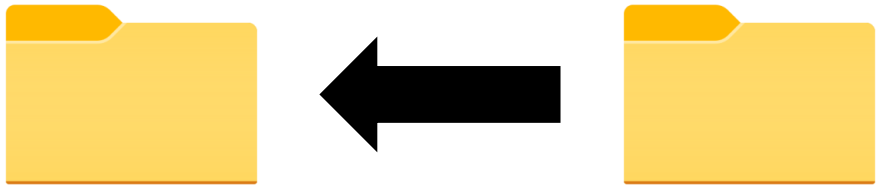
My Project

```
$ git init
Initialized empty Git repository in
C:/Users/Mikkel/Documents/GitHub crash course/.git/
```

Start a new project…

- Create a folder at the desired location
- Go to folder
- Initialize folder
- Set remote

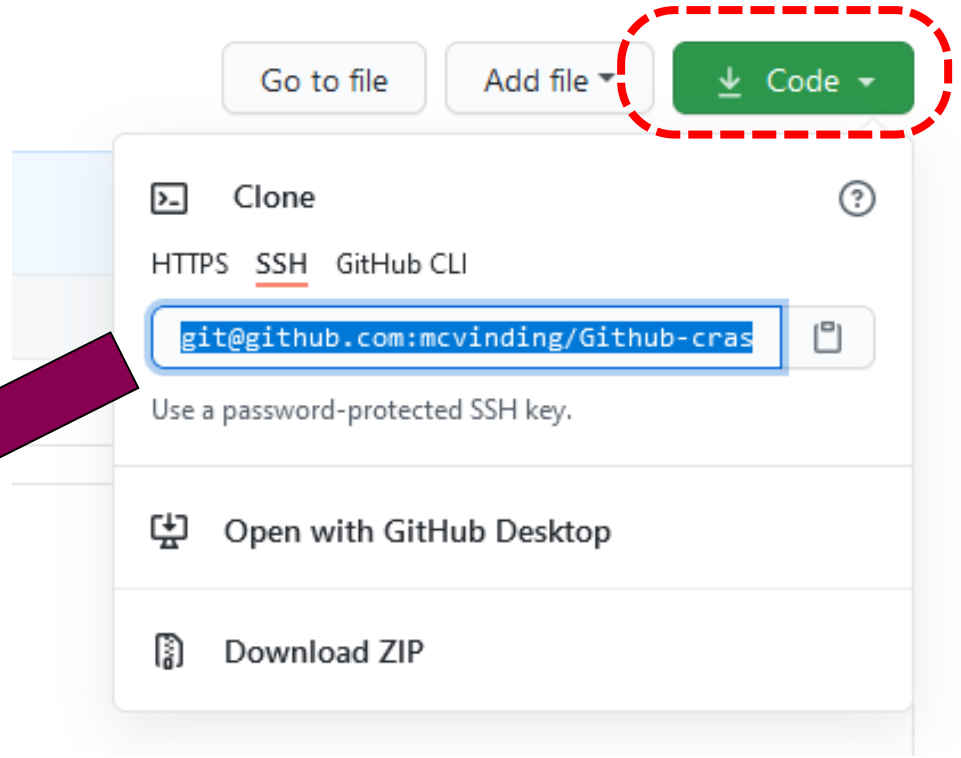
GitHub



Start a new project…

- Create a folder at the desired location
- Go to folder
- Initialize folder
- Set remote

```
$ git remote add origin <address>
```



Start a new project…

- Create a folder at the desired location
- Go to folder
- Initialize folder
- Set remote

GitHub

```
$ git remote add origin <address>
```

My Project
“origin”

My Project

See remote address

```
$ git remote -v
origin  git@GitHub.com:mcvinding/GitHub-crash-course.git (fetch)
origin  git@GitHub.com:mcvinding/GitHub-crash-course.git (push)
```

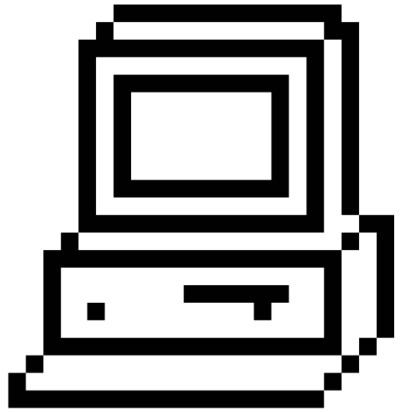
Start a new project…

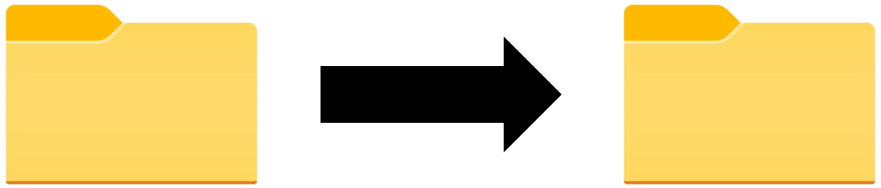
- Create a folder at the desired location
- Go to folder
- Initialize folder
- Set remote
- Get files from remote: ”pull”

```
$ git pull origin main
```

”Branch”

GitHub





My Project
“origin”

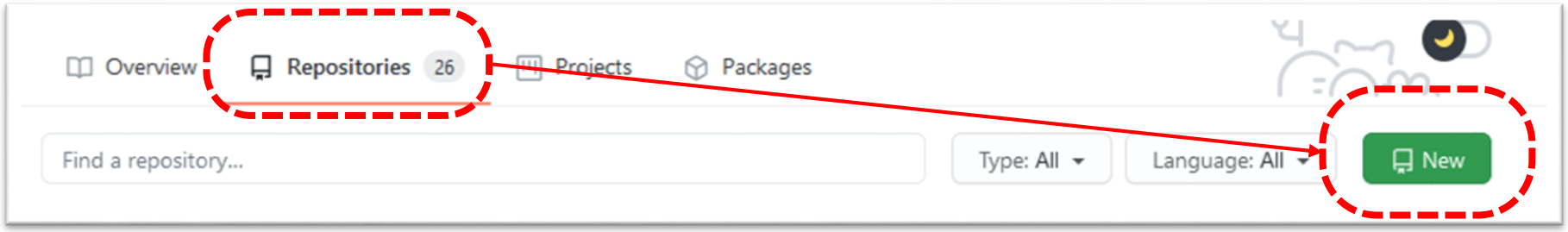
My Project

Pull files from remote

```
$ git pull origin main
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
From GitHub.com:mcvinding/GitHub-crash-course
 * branch            main       -> FETCH_HEAD
```

Summary: initiate a Git project

GitHub



```
$ git init

$ git remote add origin <address>

$ git pull origin main
```

Getting started…

Start a new project

Create a repository for your project.

Where you want to start!

Join a project

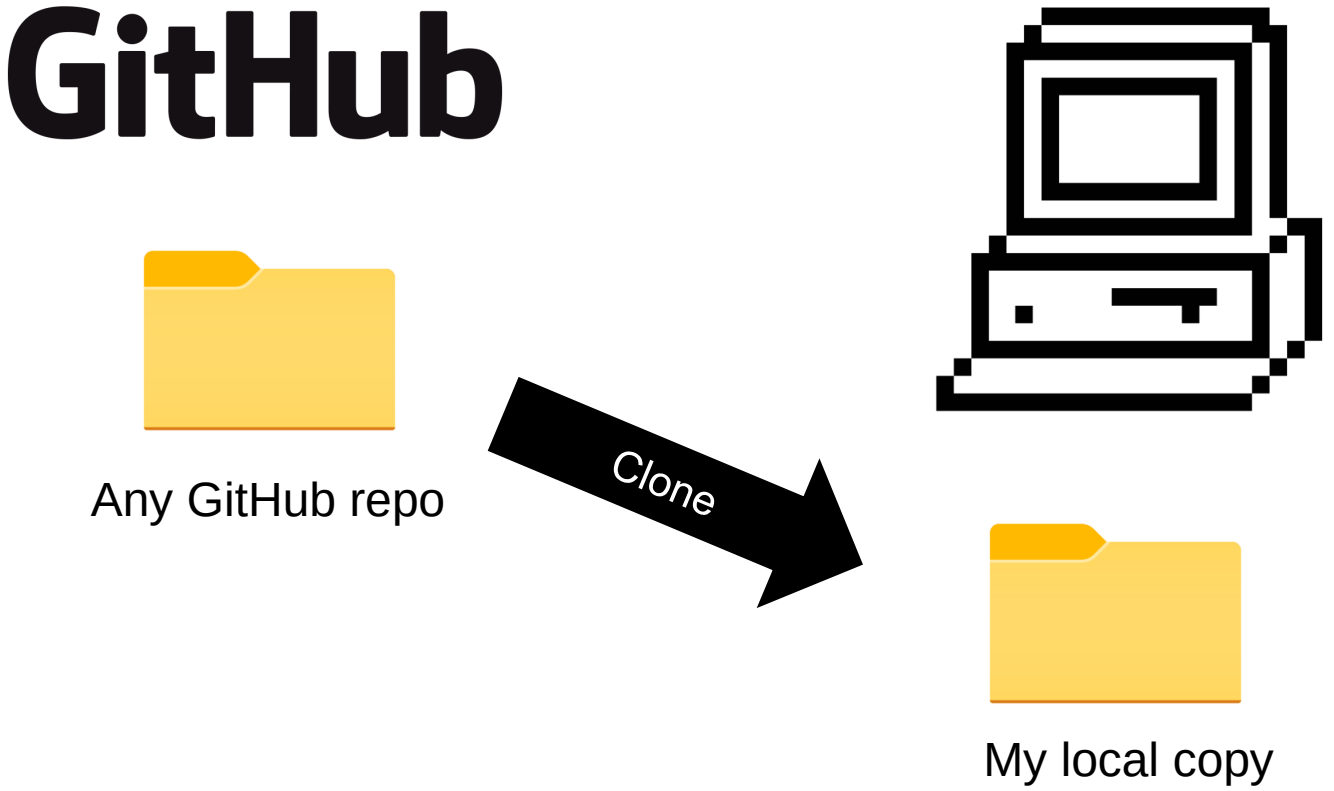
E.g. a project where someone already hace created a repository

Clone a project

Get code from another project, e.g. an analysis toolbox

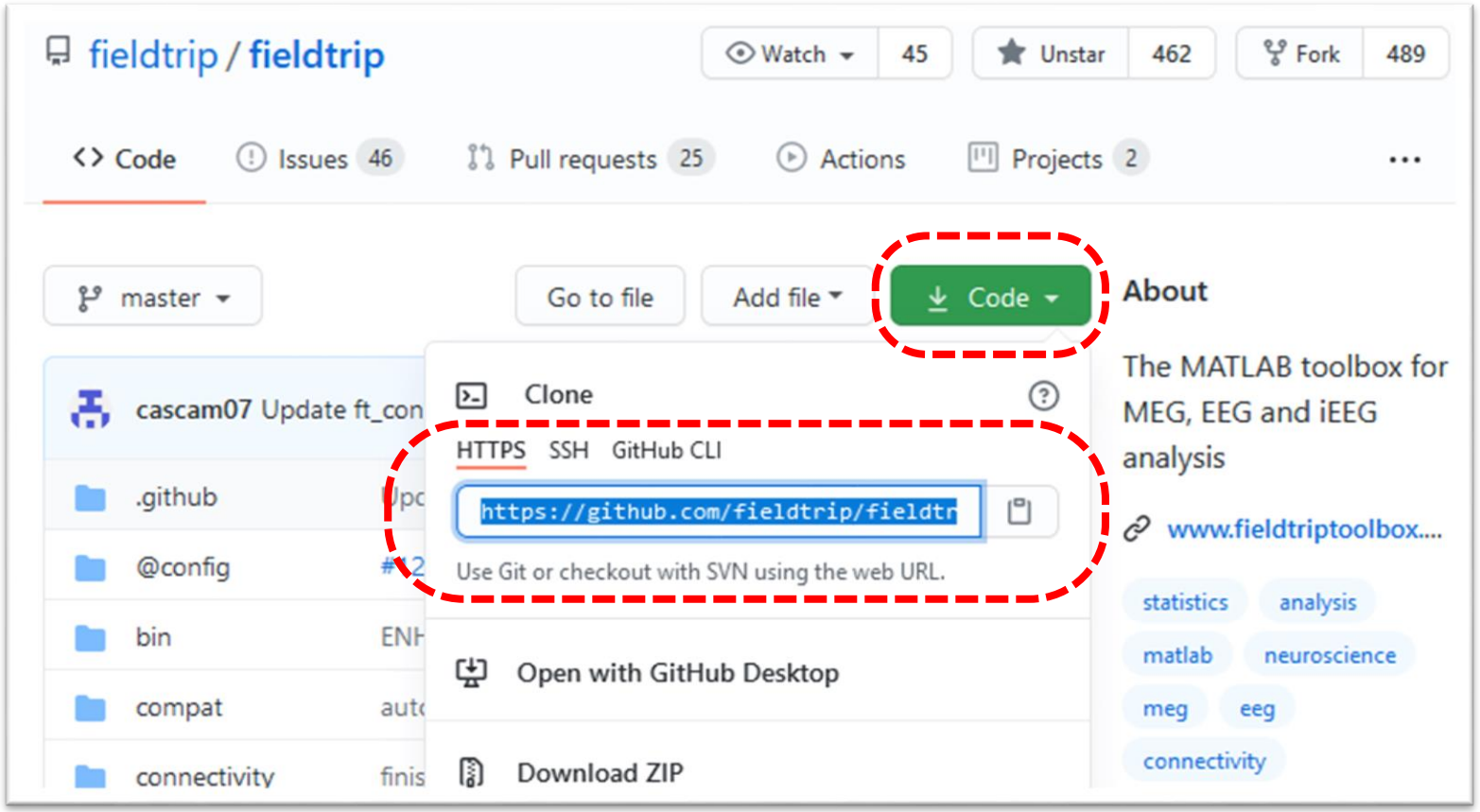
Git clone

```
$ git clone <address>
```

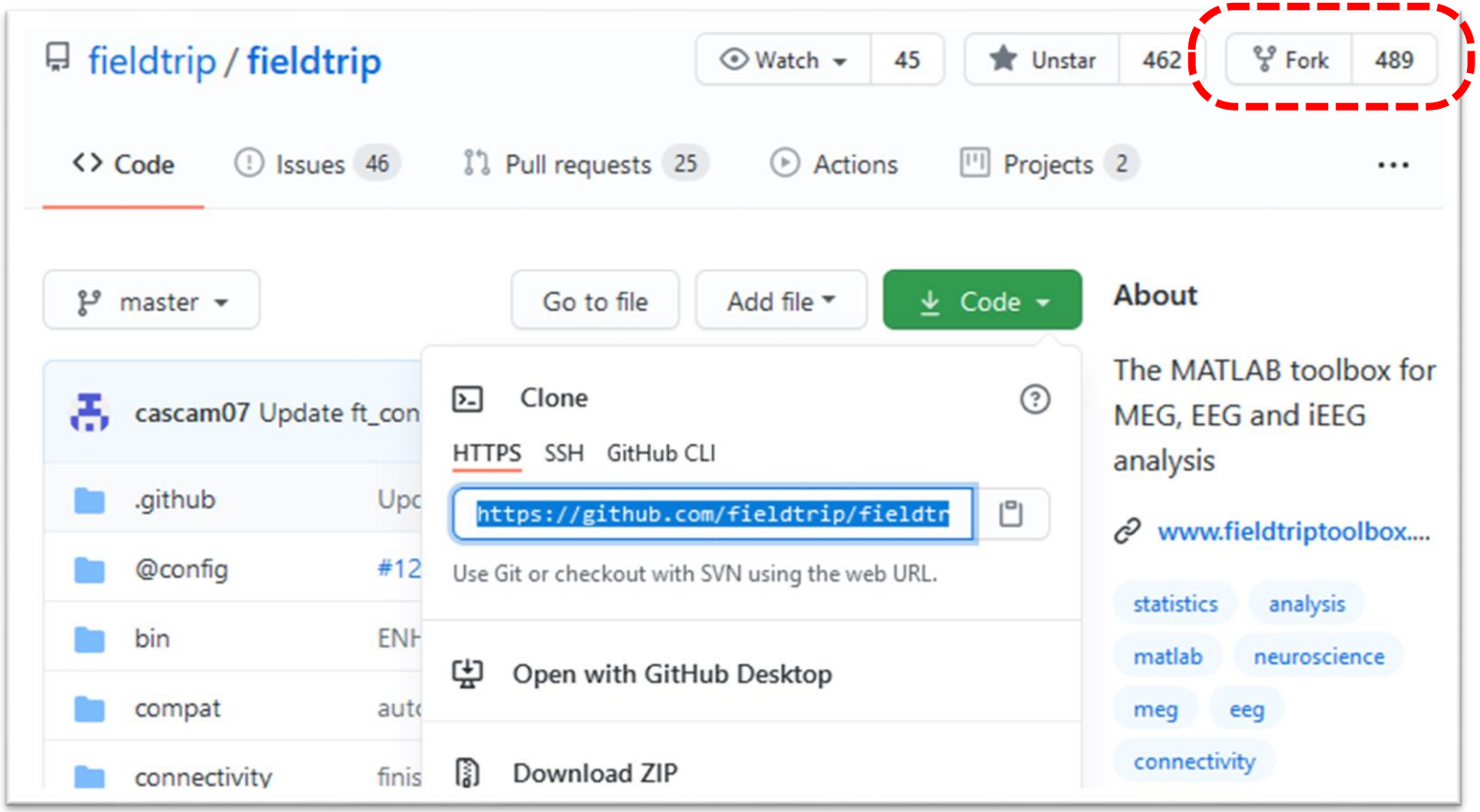


Git clone

```
$ git clone <address>
```



Fork

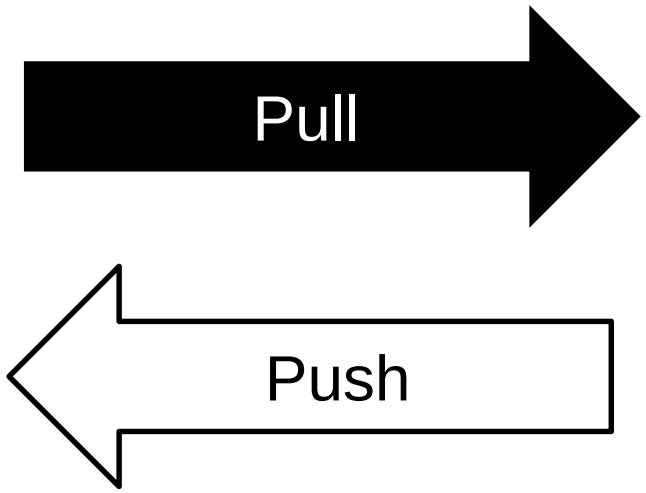

fieldtrip / fieldtrip
Watch 45
Unstar 462
Fork 489
Code
Issues 46
Pull requests 25
Actions
Projects 2
master
Go to file
Add file
Code
About
The MATLAB toolbox for MEG, EEG and iEEG analysis
www.fieldtriptoolbox....
statistics
analysis
matlab
neuroscience
meg
eeg
connectivity
cascam07 Update ft_con
.github
@config
bin
compat
connectivity
Clone
HTTPS
SSH
GitHub CLI
https://github.com/fieldtrip/fieldtr
Use Git or checkout with SVN using the web URL.
Open with GitHub Desktop
Download ZIP

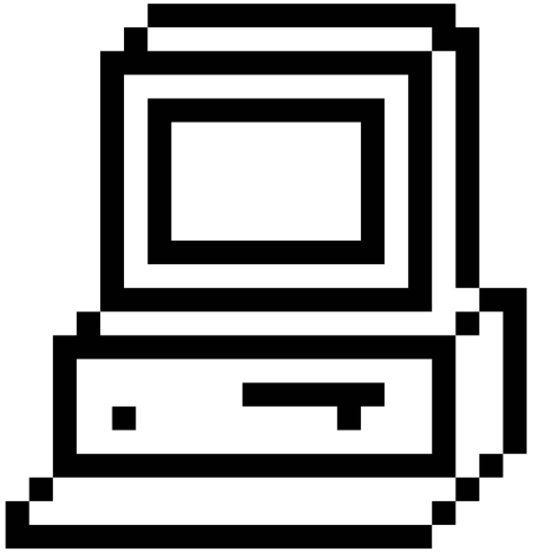
Pull, commit, push, and win at code management

WORKING WITH GIT REPOSITORIES

Git terminology

GitHub



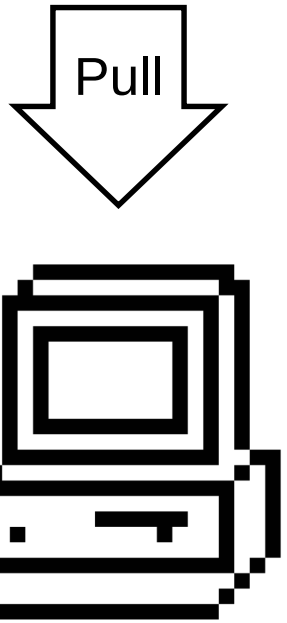


Git workflow at a glance

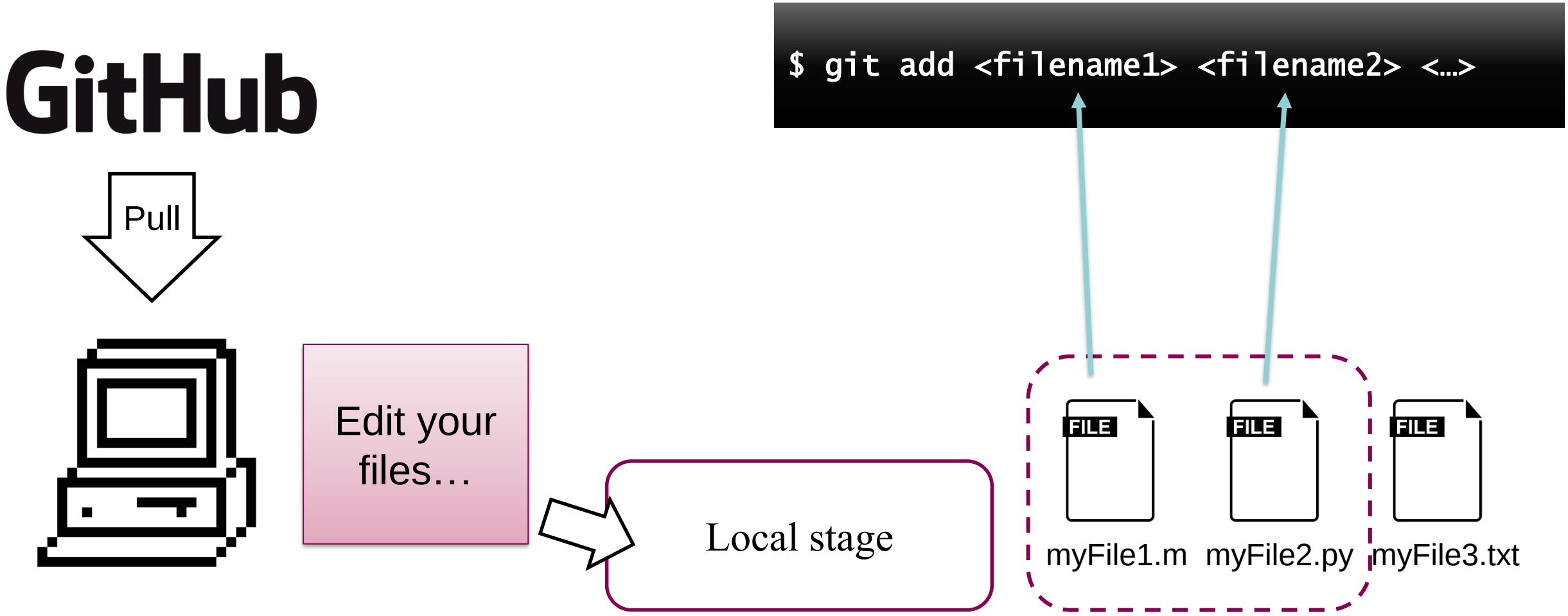
- **Pull** latest code
- Write your code…
- Stage edits
- Commit changes
- **Push**

Karolinska Institutet

GitHub




```
$ git pull <remote> <branch>
```

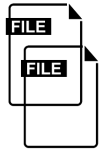

GitHub
$ git add <filename1> <filename2> <…>
Pull
Edit your files…
Local stage
FILE
FILE
FILE
myFile1.m
myFile2.py
myFile3.txt

The most used Git command

```
$ git status
```

```
$ git status
On branch main
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   new_file.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be
committed)
  (use "git checkout -- <file>..." to discard changes
in working directory)

        modified:   another_new_file.txt

Untracked files:
  (use "git add <file>..." to include in what will be
committed)

        images/
```

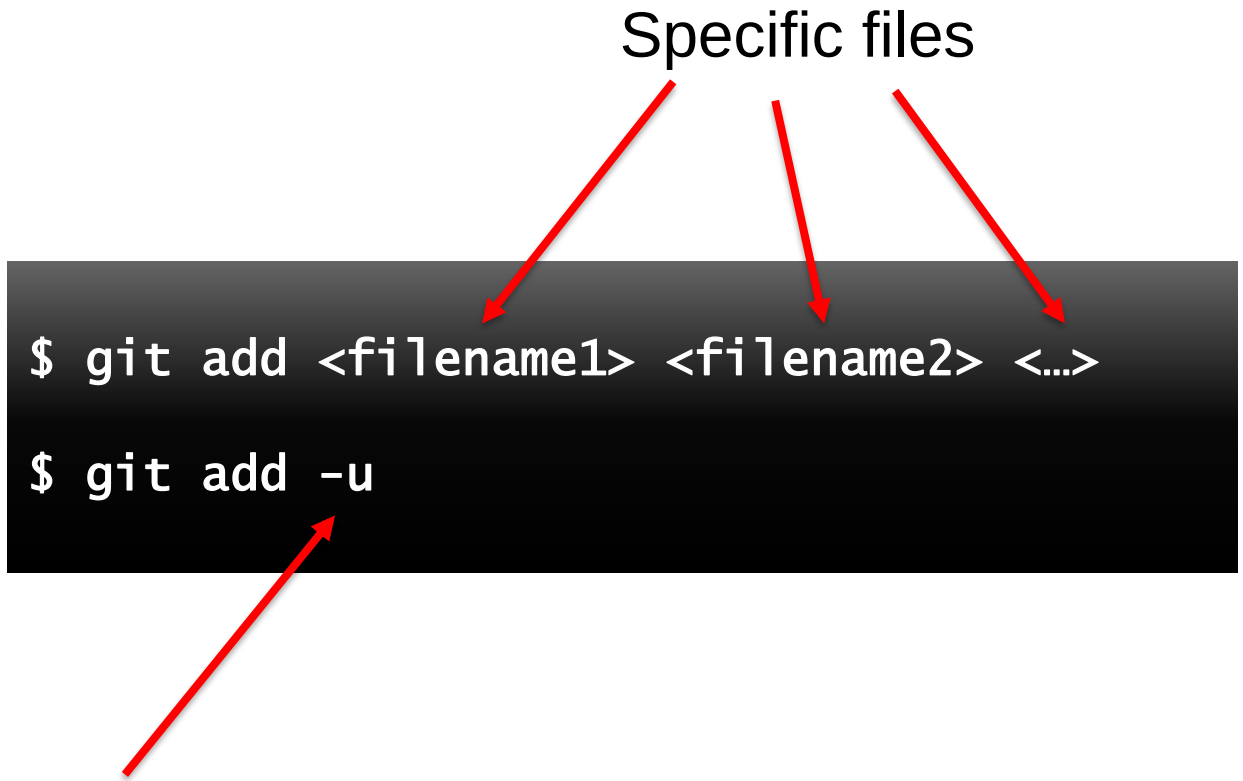
Local stage



”Off stage”

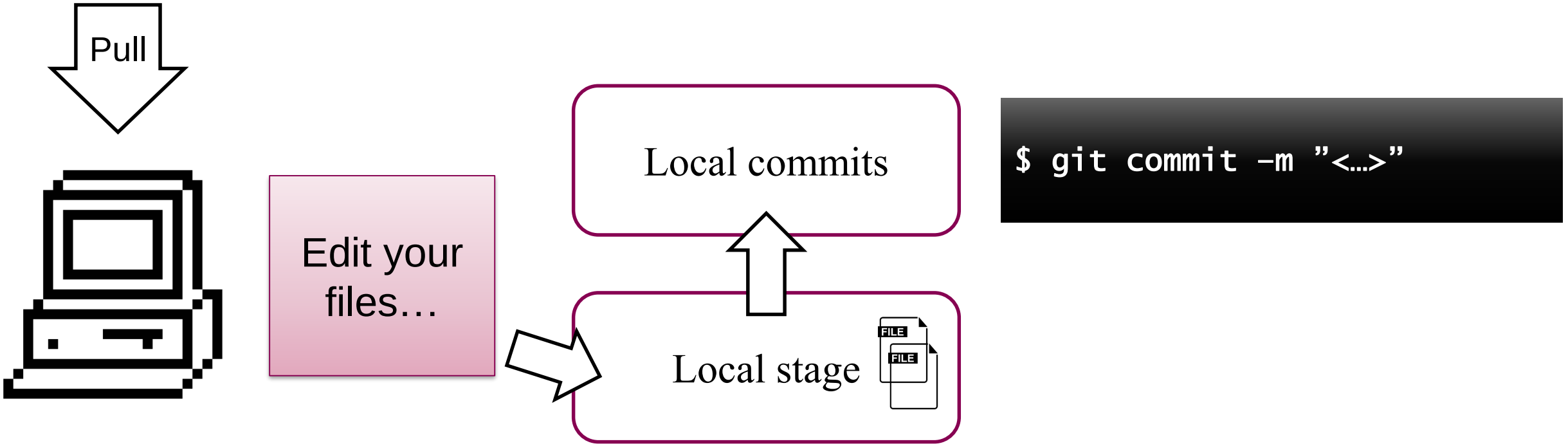
”Not even…”

Staging files



Karolinska Institutet

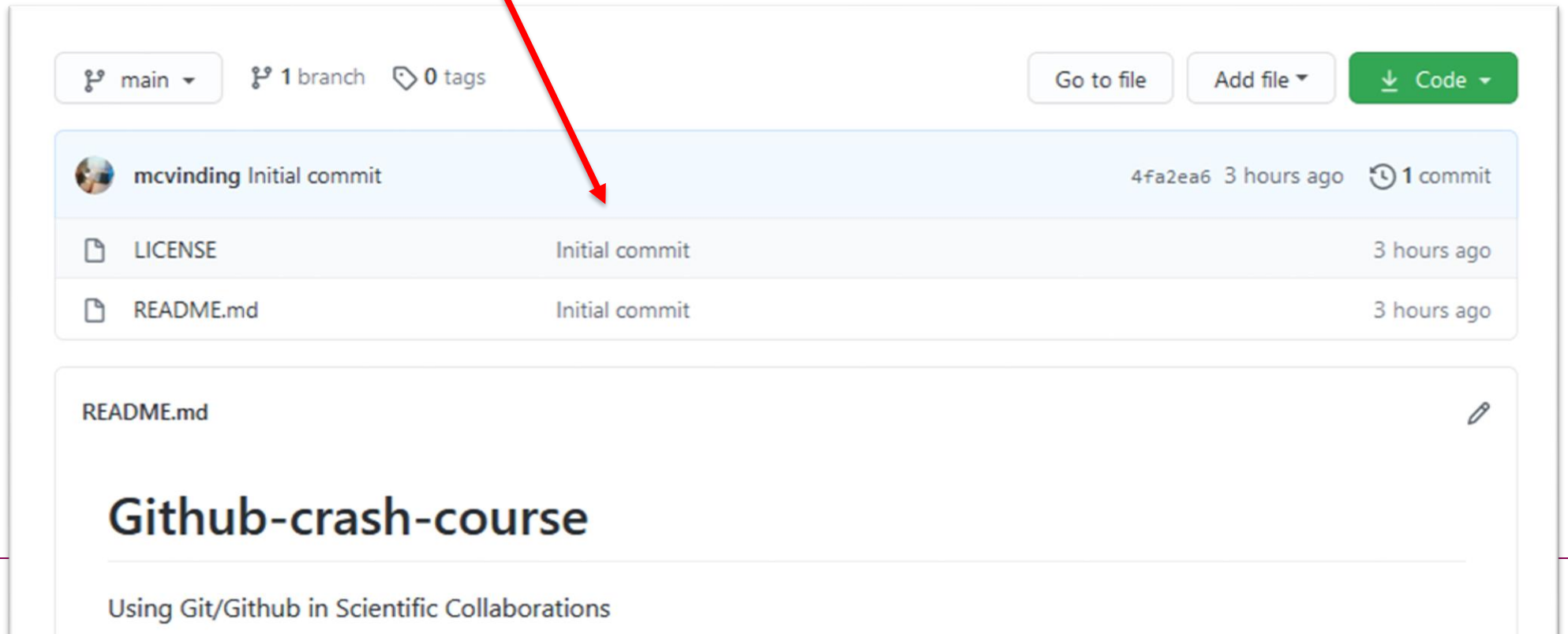
GitHub



Commit messages

```
$ git commit -m "<your message>"
```

Write a short informative text about changes

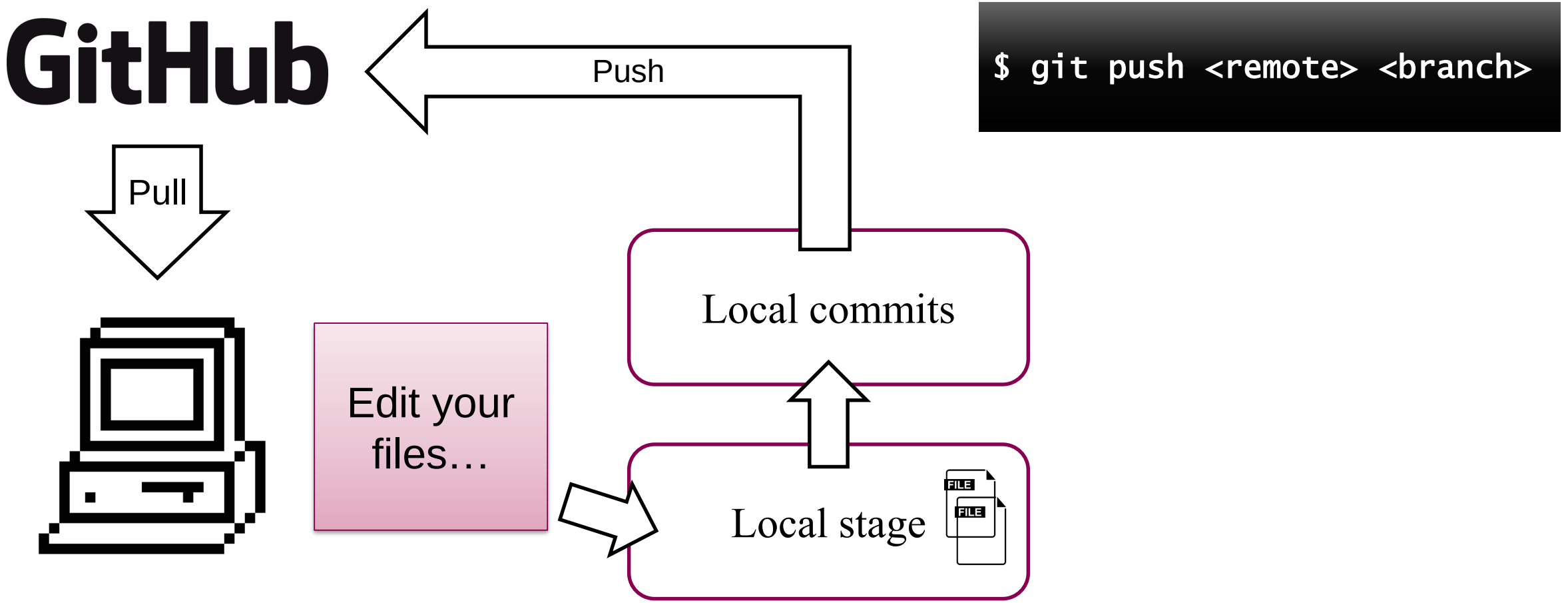


Commit messages

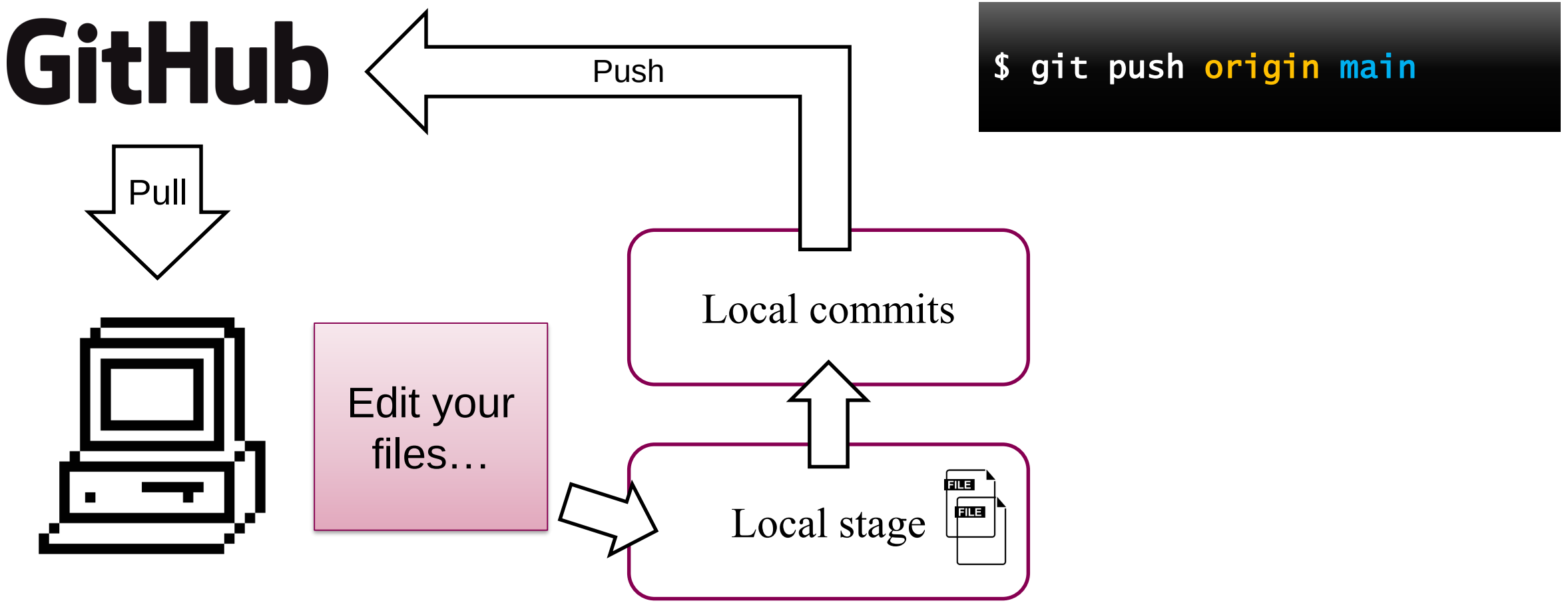
```
$ git commit -m ”<your message>”
```

Write a short informative text about changes

Too little	OK	Too much
”Some changes”	*”changed filter settings”*	*”changed filter settings in line 42-46 and added a bunch more options for processing and new scipts for plots with pretty colours that look like a pretty flower”*
”Committed stuff”	*”added files for plots”*	
	”overhaul of preprocessing”	

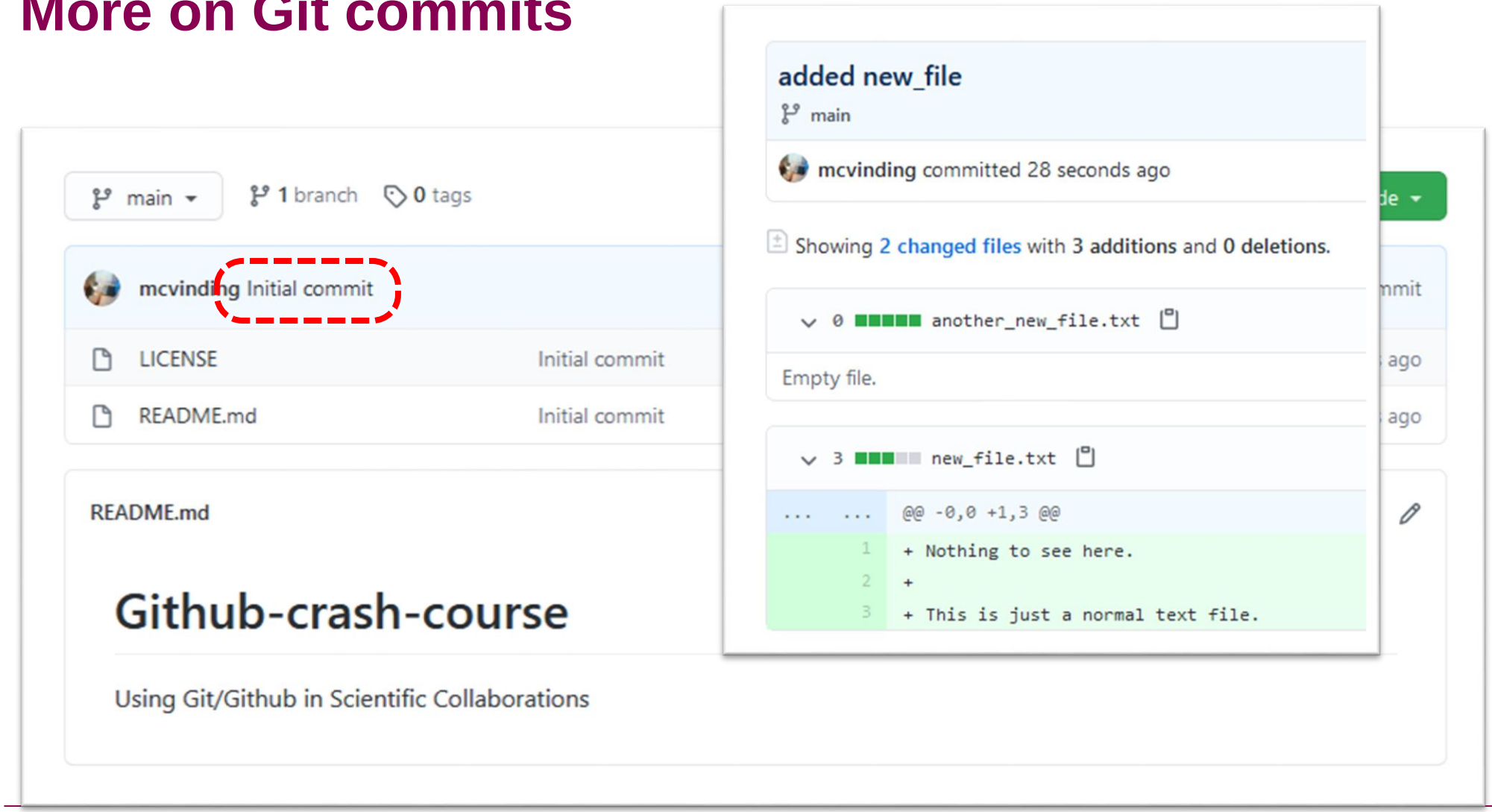



```
$ git push <remote> <branch>
```




```
$ git push origin main
```

More on Git commits



Git workflow

- **Pull** latest code
- Write your code…
- Stage edits (**add**)
- **Commit** changes
- **Push**

```
$ git pull origin main

$ git add <options>

$ git commit –m ”…”

$ git push <remote> <branch>
```

Karolinska Institutet

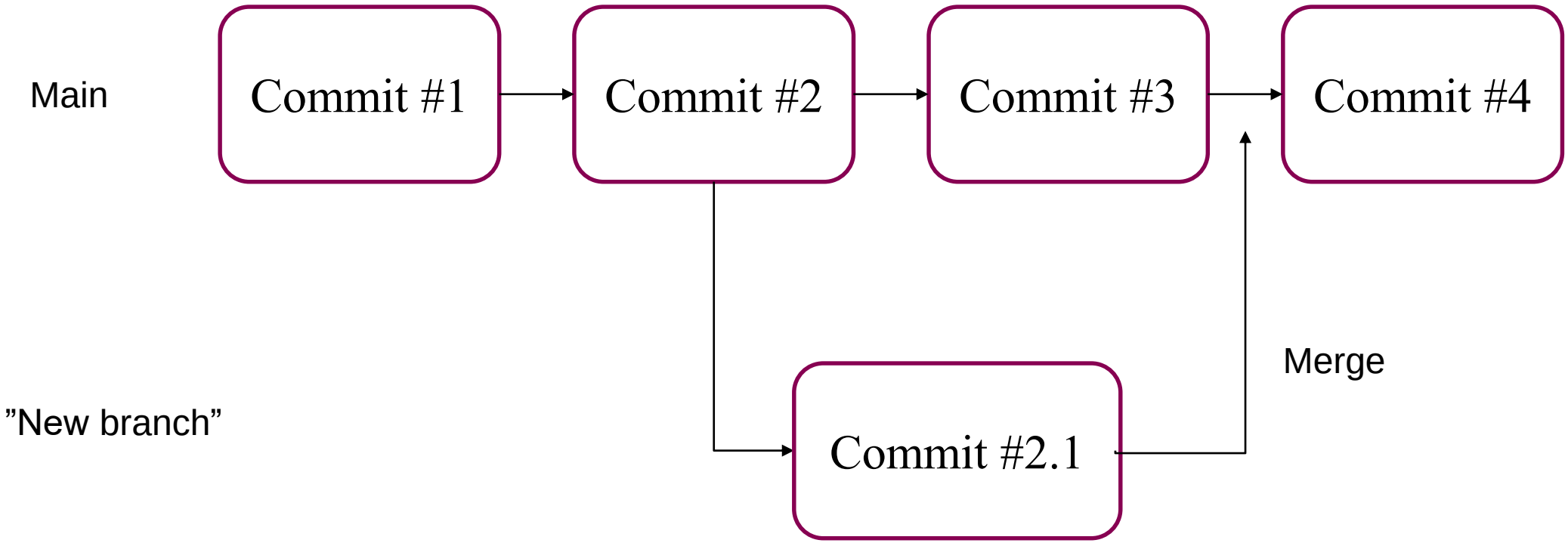
How often should I commit?

How often you want

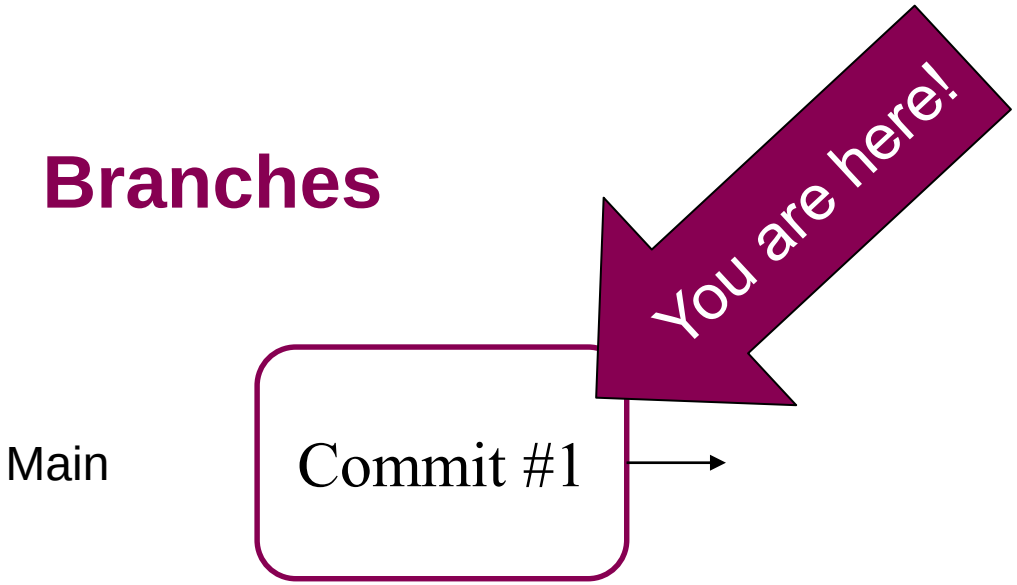
Karolinska Institutet

GIT BRANCHES

Branches

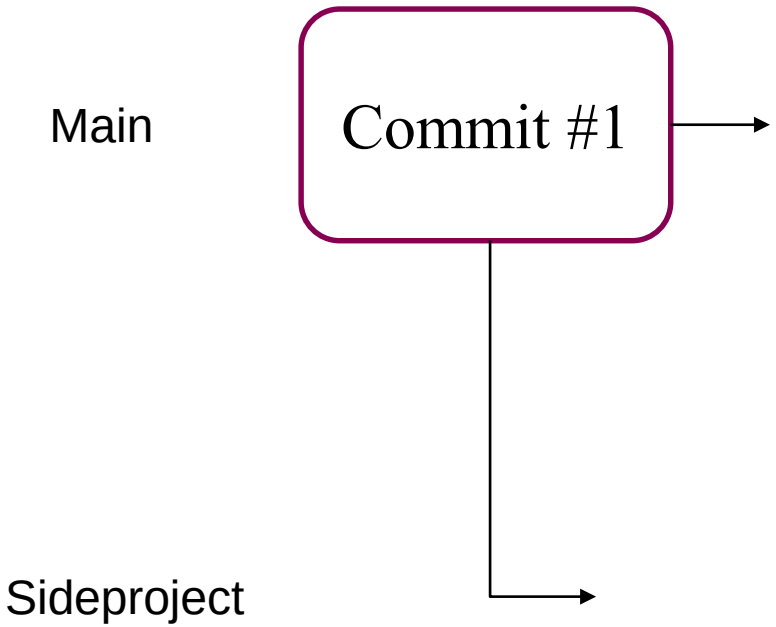

Main
Commit #1
Commit #2
Commit #3
Commit #4
”New branch”
Commit #2.1
Merge

Branches




```
$ git branch
*  Main
```

Branches

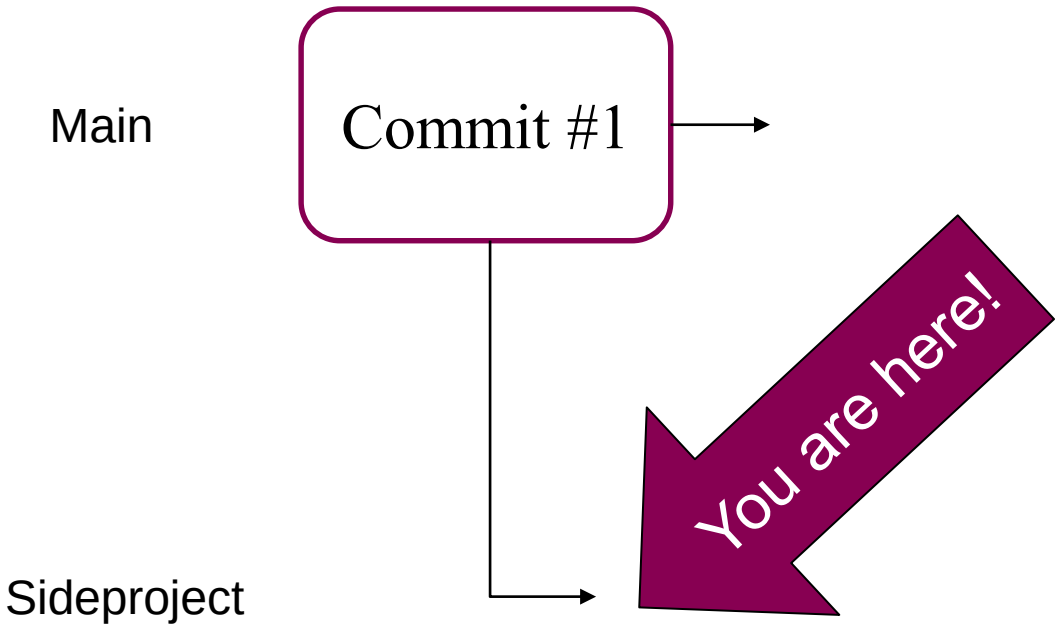



```
$ git branch
*  Main

$ git checkout -b sideproject
```

Create new flag

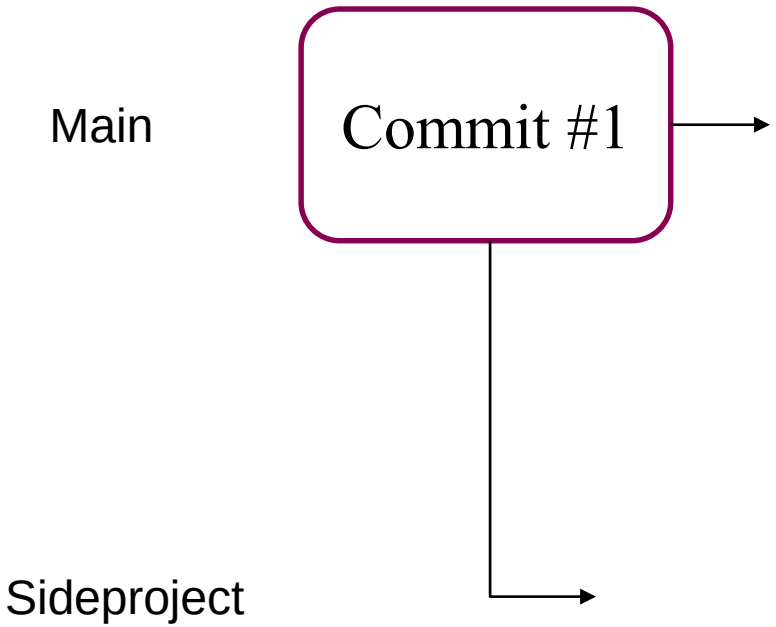
Name of new branch

Branches



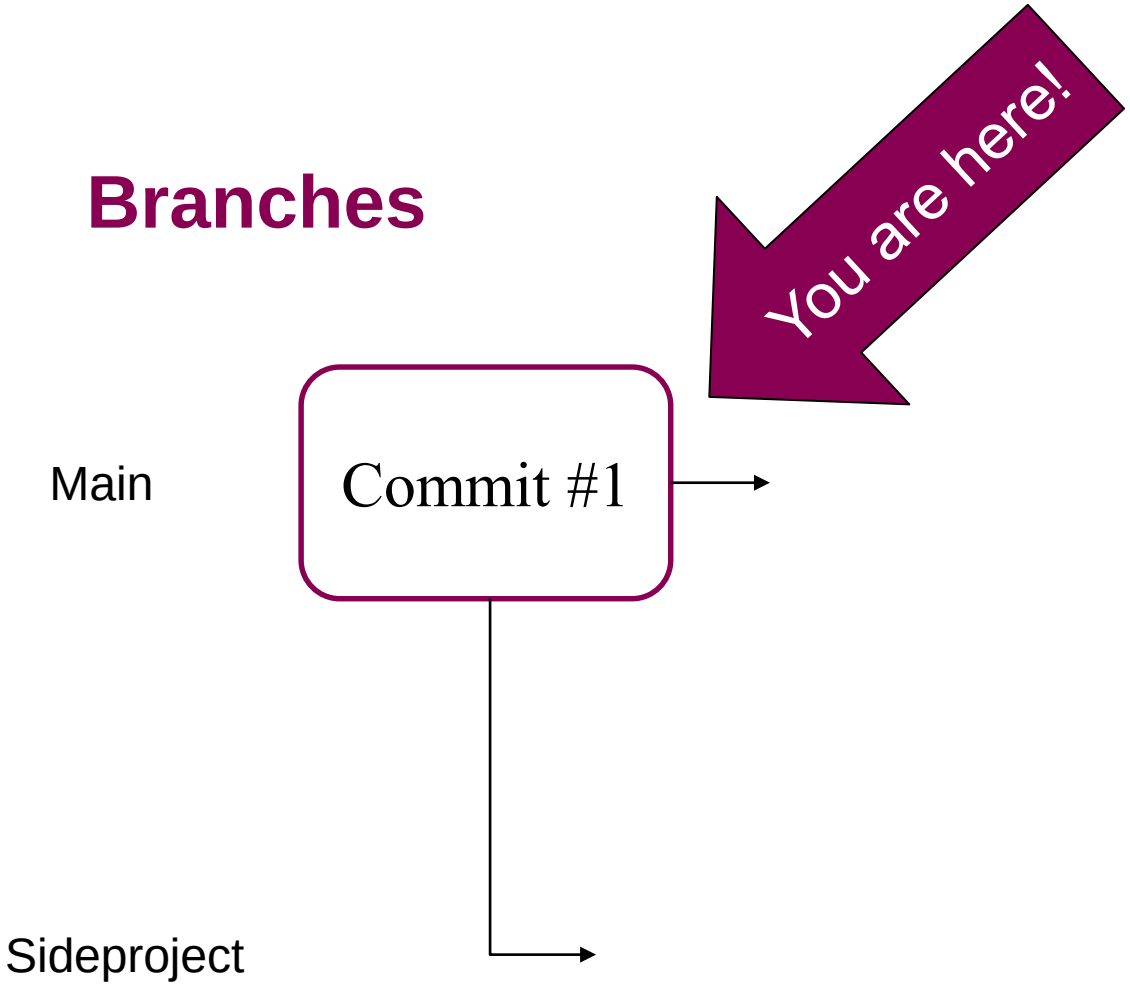

```
$ git branch
*  Main

$ git checkout -b sideproject

$ git branch
  main
* sideproject
```

Branches




```
$ git commit -m ”<…>”

$ git push origin sideproject
```

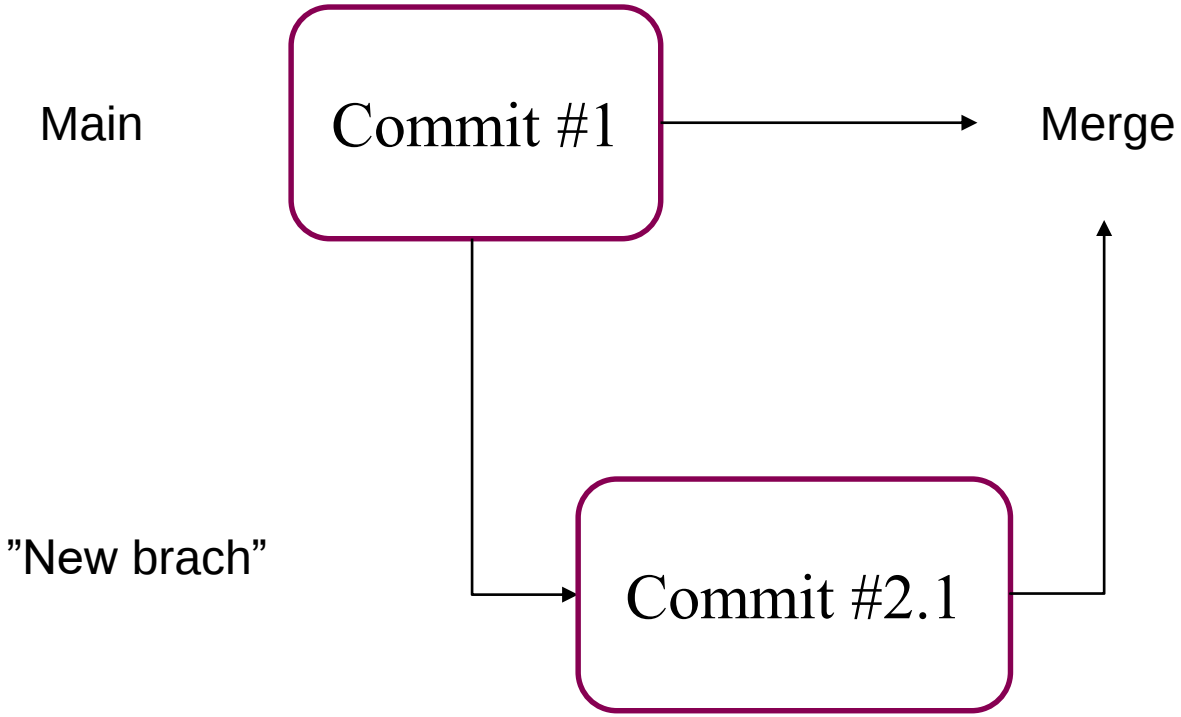
Branches



No ”-b” flag

```
$ git checkout main

$ git branch
*  main
   sideproject
```

Merge

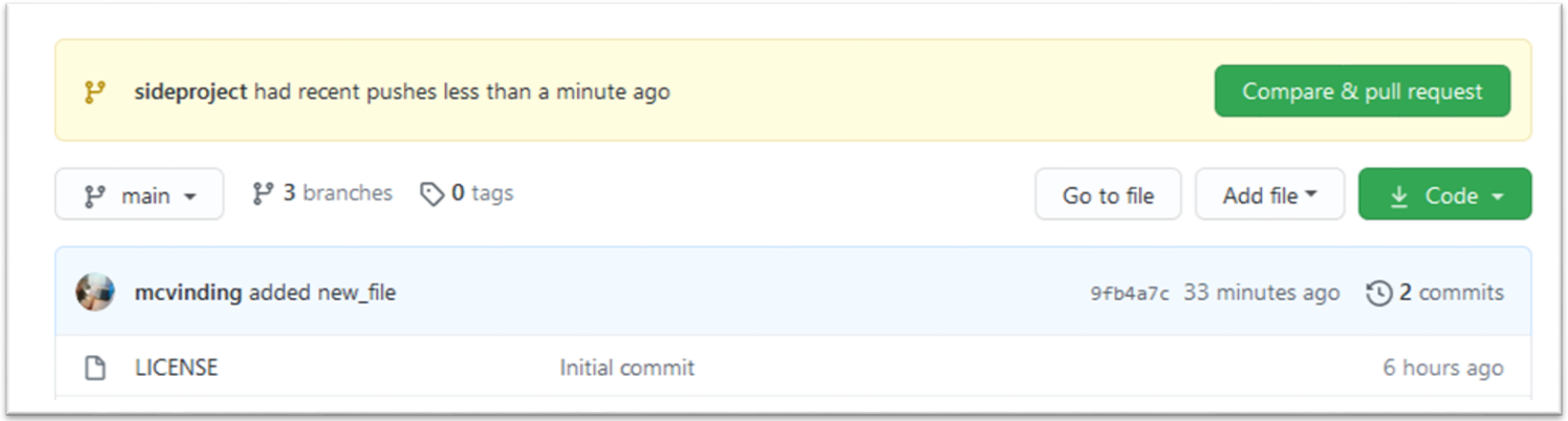


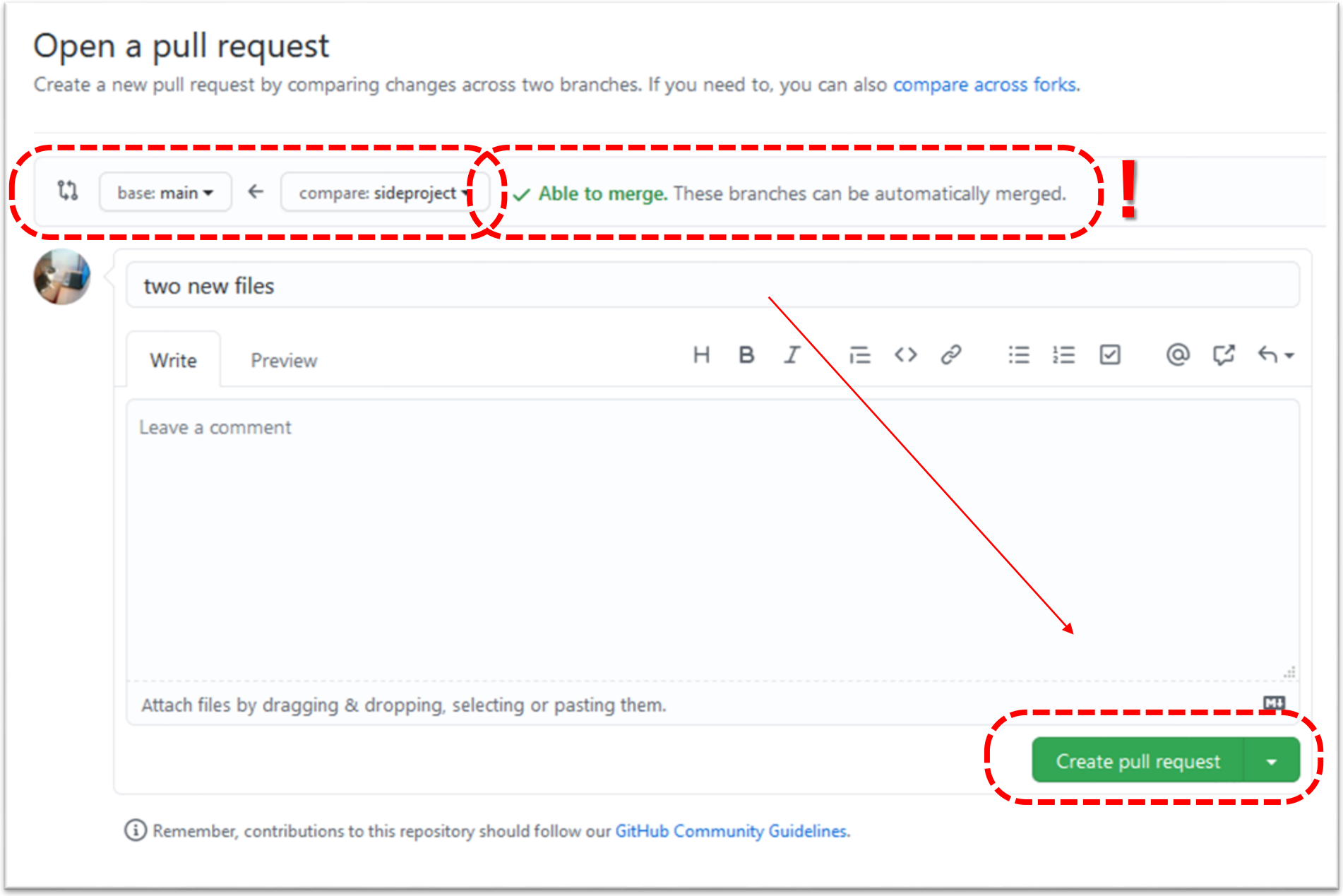

```
$ git checkout main

$ git merge sideproject
```

Alternative merge

```
$ git push origin sideproject
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'sideproject' on GitHub by visiting:
remote:      https://GitHub.com/mcvinding/GitHub-crash-
course/pull/new/sideproject
remote:
To GitHub.com:mcvinding/GitHub-crash-course.git
 * [new branch]      sideproject -> sideproject
```

Alternative merge




Open a pull request
Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.
base: main
compare: sideproject
Able to merge. These branches can be automatically merged.
two new files
Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Create pull request
Remember, contributions to this repository should follow our GitHub Community Guidelines.

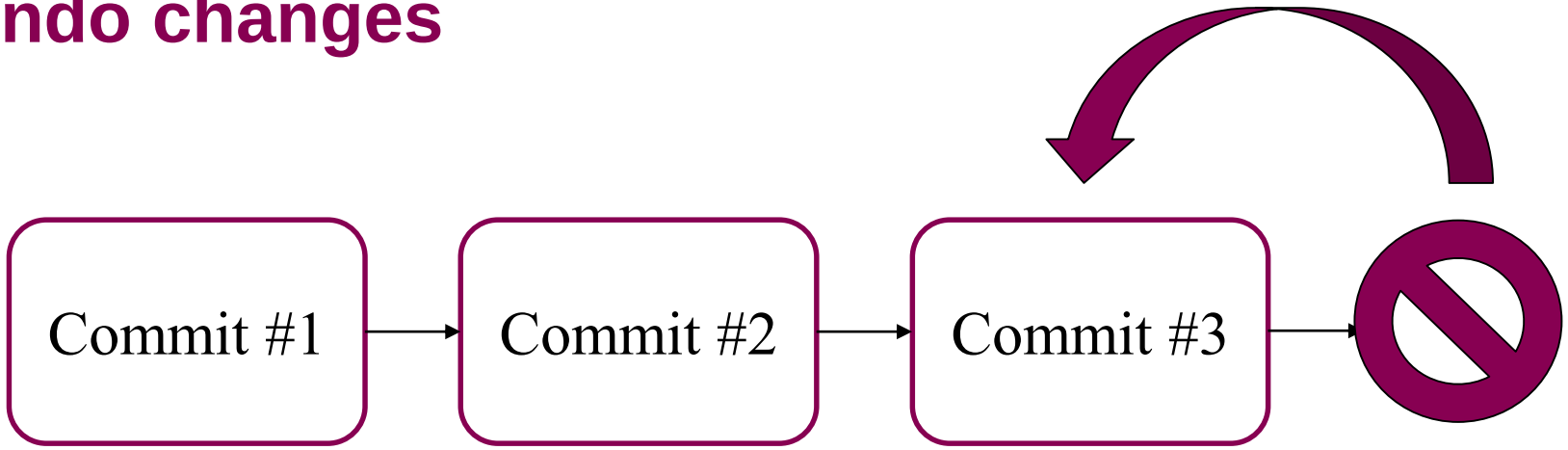
When to use branches

- When implementing entirely new features/analysis parts
- Testing out new code (and don't want to risk ruining the old)
- More useful when collaborating and delegating tasks

When not to use branches (stay on main)

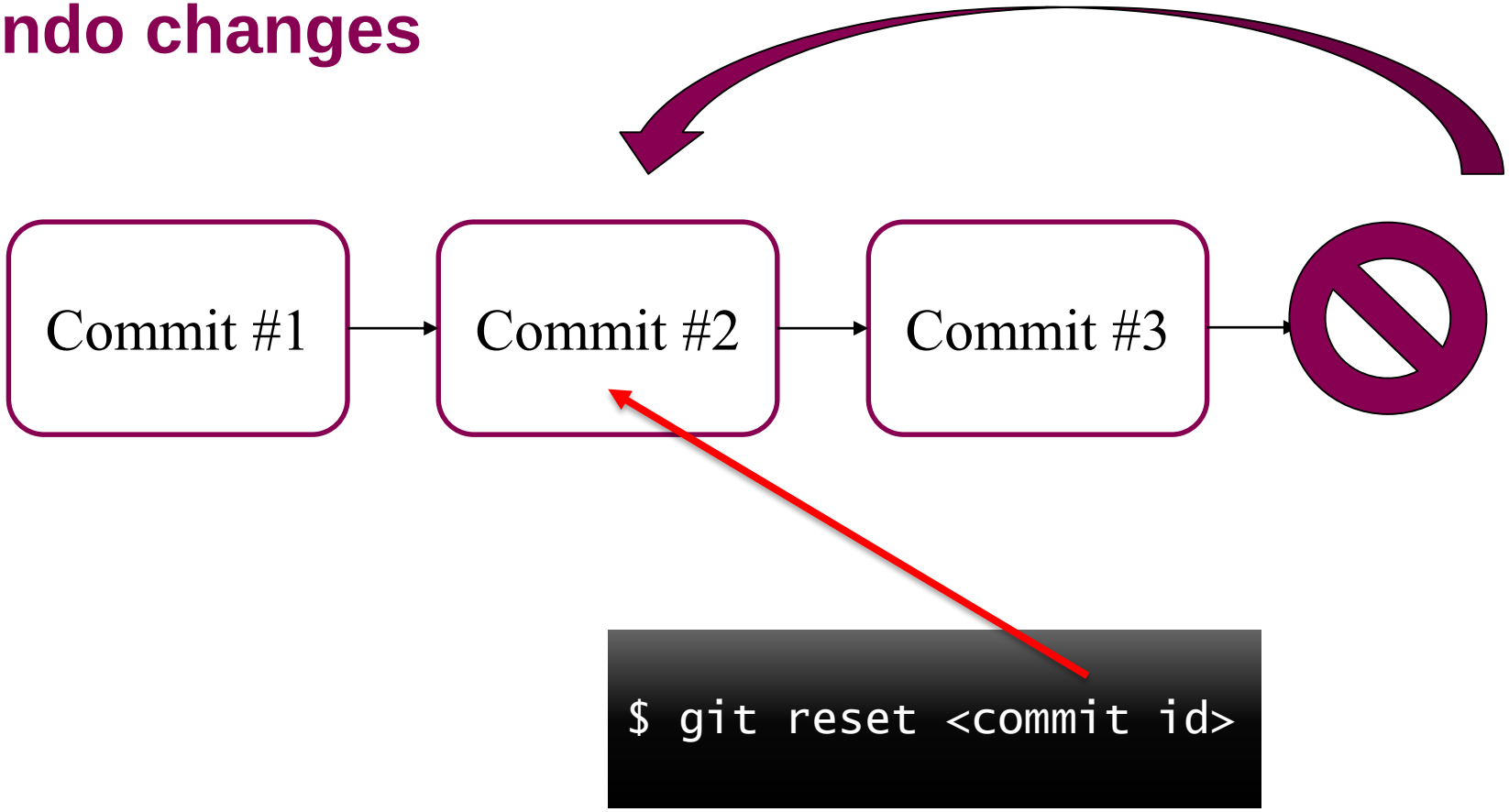
- Changes you will do anyway
- To scrap the current project

Undo changes

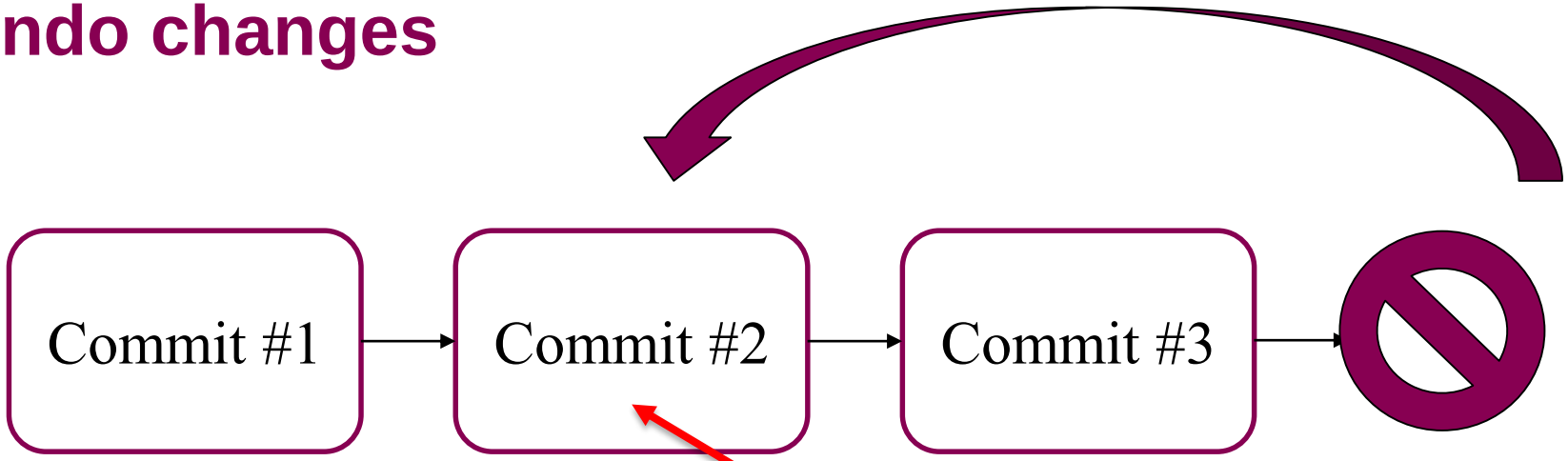



```
$ git reset --hard HEAD
```

Undo changes



Undo changes



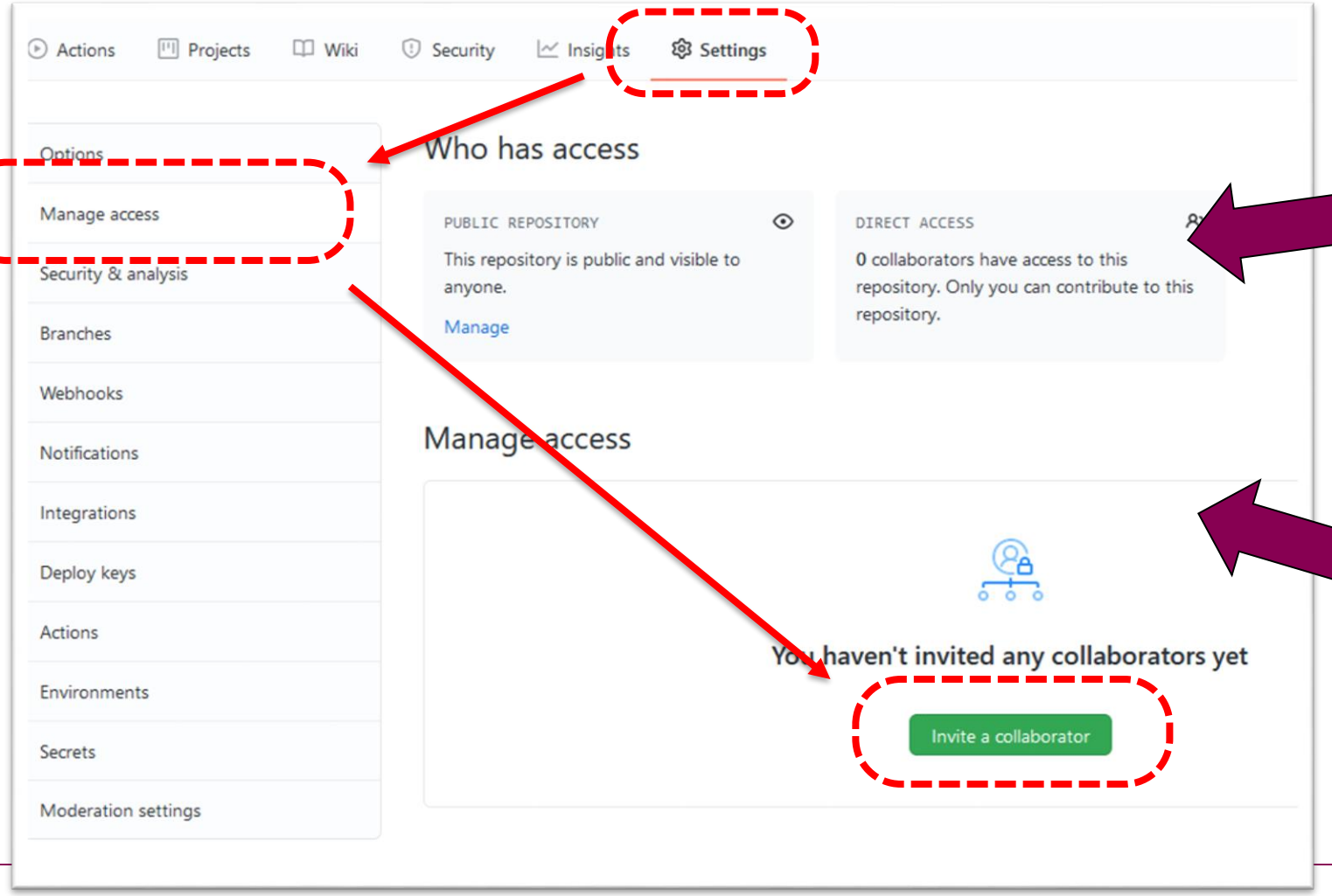

```
$ git reset 4fa2ea669e898f2ba0549f50eada713ccd65cf89
```

```
$ git log
commit 9fb4a7cbe74a87abccb5fd3fb1d04e529f3417bb (HEAD ->
main, origin/main)
Author: mcvinding <mikkel.vinding@gmail.com>
Date:   Mon Jan 20 16:33:56 2021 +0100

    added new_file

commit 4fa2ea669e898f2ba0549f50eada713ccd65cf89
(origin/master, main)
Author: MC Vinding <mikkel.vinding@gmail.com>
Date:   Mon Jan 20 16:26:53 2021 +0100

    Initial commit
```

The more, the merrier: using GitHub with your lab mates

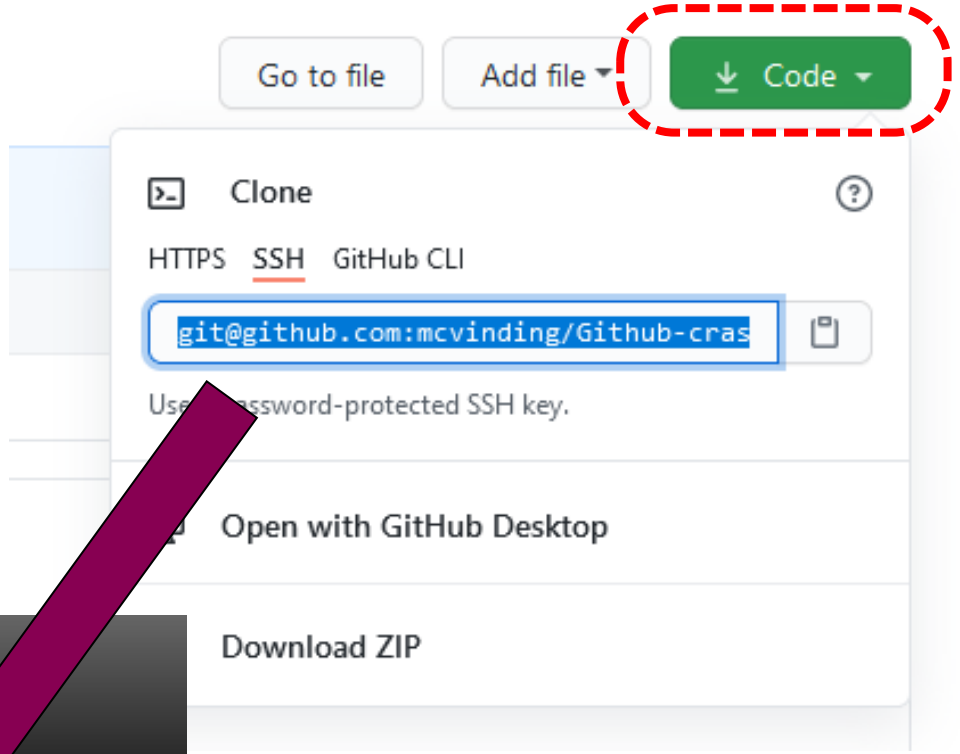
GITHUB COLLABORATIONS

Adding collaborators



Setting up collaboration

- Initialize folder on your computer
- Go to your collaborators project
- Add remote collaborators remote



```
$ git init

$ git remote add origin <address>

$ git pull origin main
```

Collaborating (same procedure as before)

- **Pull** latest code
- Write your code…
- Stage edits
- Commit changes
- **Push**

```
$ git pull origin main

$ git add <options>

$ git commit

$ git push <remote> <branch>
```

Summary

- Use Git/GitHub to manage your analysis scripts (not data).
- Basic usage (same if you are working on your own or in collaboration)

```
$ git remote add <remote> <branch>

$ git pull origin main

$ git add <options>

$ git commit

$ git push <remote> <branch>
```